МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: **«СТВОРЕННЯ ВЕБЗАСТОСУНКУ ДЛЯ ПЕРСОНАЛІЗОВАНОГО НАВЧАННЯ ІНОЗЕМНИХ МОВ ІЗ ЗАСТОСУВАННЯМ АДАПТИВНИХ ТЕСТІВ ТА ІГРОВИХ ЕЛЕМЕНТІВ»**

Виконав: студент 4 курсу, групи 6.1222
спеціальності 122 комп'ютерні науки
(шифр і назва спеціальності)
освітньої програми комп'ютерні науки
(назва освітньої програми)
Д. Е. Каськов
(ініціали та прізвище)
Керівник доцент кафедри комп'ютерних наук, к.т.н. Добровольський Г.А.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
Рецензент доцент кафедри програмної інженерії, к.ф.-м.н. Мильцев О. М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя 2026

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти бакалавр

Спеціальність 122 комп'ютерні науки
(шифр і назва)

Освітня програма комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри комп'ютерних наук, к.пед.н., доцент

_____ Пшенична О.С.
(підпис)

" 08 " грудня 2025 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Каськову Данилу Едуардовичу
(прізвище, ім'я та по-батькові)

1. Тема роботи Створення вебзастосунку для персоналізованого навчання іноземних мов із застосуванням адаптивних тестів та ігрових елементів

керівник роботи Добровольський Геннадій Анатолійович, к.т.н., доцент
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 02 » грудня 2025 року № 1741-с

2. Строк подання студентом роботи 04.06.2026

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Теоретичні основи створення вебзастосунків для персоналізованого навчання іноземних мов.
2. Проєктування вебзастосунку для персоналізованого навчання іноземних мов.
3. Розробка та тестування вебзастосунку.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 05.12.2026

К А Л Е Н Д А Р Н И Й П Л А Н

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	01.03.2026	
2.	Збір вихідних даних.	22.03.2026	
3.	Обробка методичних та теоретичних	05.04.2025	
	джерел.		
4.	Розробка першого та другого розділу.	27.04.2026	
5.	Розробка третього розділу.	19.05.2026	
6.	Оформлення та нормоконтроль	04.06.2026	
	кваліфікаційної роботи бакалавра.		
7.	Захист кваліфікаційної роботи.	19.06.2026	

Студент _____ Д. Е. Каськов
(підпис) (ініціали та прізвище)

Керівник роботи _____ Г. А. Добровольський
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ О.Г.Спиця
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Створення вебзастосунку для персоналізованого навчання іноземних мов із застосуванням адаптивних тестів та ігрових елементів»: 74 с., 15 рис., 11 табл., 29 джерел, 2 додатків.

АДАПТИВНЕ НАВЧАННЯ, ВЕБЗАСТОСУНОК, ГЕЙМІФІКАЦІЯ, ІНДИВІДУАЛЬНЕ НАВЧАННЯ, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС, ОСВІТНІ ТЕХНОЛОГІЇ, ПЕРСОНАЛІЗОВАНЕ НАВЧАННЯ, ТЕСТИ, LMS.

Об’єктом дослідження є процес навчання іноземних мов із використанням цифрових технологій.

Предметом дослідження є методи та засоби розробки вебзастосунків для персоналізованого навчання іноземних мов.

Метою роботи є розробка вебзастосунку для персоналізованого навчання іноземних мов із застосуванням адаптивних тестів та елементів гейміфікації, що забезпечує індивідуалізацію навчального процесу та підвищення ефективності засвоєння матеріалу.

Методи дослідження включають аналіз і узагальнення наукових джерел, системний аналіз, проєктування інформаційних систем, моделювання структури баз даних, об’єктно-орієнтоване програмування, а також функціональне й інтеграційне тестування програмного забезпечення.

У роботі проаналізовано сучасні підходи до персоналізованого навчання іноземних мов та існуючі освітні платформи. Спроєктовано архітектуру вебзастосунку, структуру бази даних і основні функціональні модулі. Реалізовано адаптивне тестування та систему персоналізованих рекомендацій. Впроваджено елементи гейміфікації (рівні, XP, рейтинг користувачів) для підвищення мотивації навчання. Проведене тестування підтвердило коректність роботи системи та її придатність до використання.

SUMMARY

Bachelor's Thesis «Development of a Web Application for Personalized Learning of Foreign Languages with Adaptive Tests and Game Elements»: 74 pages, 15 figures, 11 tables, 29 sources, 2 appendices.

ADAPTIVE LEARNING, WEB APPLICATION, GAMIFICATION, INDIVIDUAL LEARNING, USER INTERFACE, EDUCATIONAL TECHNOLOGIES, PERSONALIZED LEARNING, TESTS, LMS.

The object of research is the process of foreign language learning using digital technologies.

The subject of research is methods and tools for developing web applications for personalized foreign language learning.

The aim of the work is to develop a web application for personalized foreign language learning using adaptive tests and gamification elements, ensuring the individualization of the learning process and improving the effectiveness of knowledge acquisition.

Research methods include analysis and generalization of scientific sources, systems analysis, information systems design, database structure modeling, object-oriented programming, as well as functional and integration software testing.

The work analyzes modern approaches to personalized foreign language learning and existing educational platforms. The system architecture, database structure, and core functional modules are designed. Adaptive testing and personalized recommendation mechanisms are implemented. Gamification elements (levels, XP, user ranking) are introduced to increase learning motivation. Testing confirmed the correctness of the system's operation and its suitability for practical use.

ЗМІСТ

Завдання на кваліфікаційну роботу……..2

Реферат 4

Summary 5

Вступ..... 7

1 Теоретичні основи створення вебзастосунків для персоналізованого навчання іноземних мов 10

1.1 Особливості використання інформаційних технологій у вивченні іноземних мов 10

1.2 Аналіз сучасних платформ для вивчення іноземних мов та їх функціональних можливостей 13

1.3 Визначення вимог до вебзастосунку персоналізованого навчання іноземних мов 18

2 Проєктування вебзастосунку для персоналізованого навчання іноземних мов 21

2.1 Аналіз предметної області та проєктування архітектури системи 21

2.2 Проєктування бази даних та функціональних модулів вебзастосунку…. 25

2.3 Проєктування механізму адаптивного тестування та формування індивідуальної траєкторії навчання 29

2.4 Проєктування системи гейміфікації та користувацького інтерфейсу 33

3 Розробка та тестування вебзастосунку 39

3.1 Реалізація програмного продукту та опис використаних технологій 39

3.2 Реалізація функціоналу адаптивних тестів і персоналізованих рекомендацій 44

3.3 Реалізація ігрових елементів та системи мотивації користувачів 49
3.4 Тестування вебзастосунку та аналіз результатів його роботи 53
3.5 Інструкція користувача..... 54
Висновки 56
Перелік посилань..... 58
Додаток А Лістинг реалізації програмного продукту 61
Додаток Б Інтерфейс вебзастосунку 71

ВСТУП

Сучасний У сучасних умовах стрімкого розвитку інформаційних технологій та цифровізації освіти особливого значення набуває використання вебзастосунків для організації навчального процесу. Одним із найбільш актуальних напрямів є створення систем для вивчення іноземних мов, які забезпечують доступність навчальних матеріалів, гнучкість у навчанні та можливість індивідуального підходу до кожного користувача. Зростання вимог до якості мовної підготовки, а також потреба у безперервному навчанні протягом життя зумовлюють необхідність розробки сучасних рішень, що поєднують адаптивні технології та мотиваційні механізми.

Традиційні методи навчання іноземних мов часто не враховують індивідуальні особливості здобувачів освіти, такі як рівень знань, швидкість засвоєння матеріалу, стиль навчання та мотивація. Це призводить до зниження ефективності навчального процесу та втрати інтересу до навчання. У зв’язку з цим особливої актуальності набуває концепція персоналізованого навчання, яка передбачає адаптацію навчального контенту та складності завдань відповідно до потреб конкретного користувача.

Важливу роль у реалізації персоналізованого навчання відіграють адаптивні вебзастосунки, які здатні аналізувати результати діяльності користувача та формувати індивідуальну траєкторію навчання. Окрему увагу в сучасних освітніх платформах приділяють використанню елементів гейміфікації, що підвищують мотивацію користувачів через систему балів, рівнів, досягнень та інших ігрових механізмів. Поєднання адаптивного тестування та гейміфікації дозволяє значно підвищити ефективність засвоєння матеріалу.

Аналіз сучасних платформ для вивчення іноземних мов (таких як Duolingo, Busuu, Memrise та інших) показує, що більшість із них вже використовують елементи персоналізації та ігрових механік. Водночас існуючі рішення мають

певні обмеження, зокрема недостатньо гнучкі алгоритми адаптації складності завдань, обмежені можливості налаштування навчальної траєкторії та часткова залежність від платних функцій. Це створює потребу у розробці нових вебзастосунків, які поєднують ефективне адаптивне тестування, глибоку персоналізацію та сучасні підходи до гейміфікації.

Актуальність теми кваліфікаційної роботи зумовлена необхідністю створення вебзастосунку для персоналізованого навчання іноземних мов, який забезпечує адаптацію навчального процесу до індивідуальних потреб користувача, автоматичне формування навчальної траєкторії та підвищення мотивації за рахунок ігрових елементів. Такий підхід дозволяє зробити процес навчання більш ефективним, цікавим та результативним.

Метою роботи є розробка вебзастосунку для персоналізованого навчання іноземних мов із застосуванням адаптивних тестів та елементів гейміфікації, що забезпечує індивідуалізацію навчального процесу та підвищення ефективності засвоєння матеріалу.

Для досягнення поставленої мети необхідно виконати такі завдання:

– проаналізувати особливості використання інформаційних технологій у навчанні іноземних мов;

– дослідити сучасні платформи для вивчення іноземних мов та їх функціональні можливості;

– визначити вимоги до вебзастосунку персоналізованого навчання;

– спроєктувати архітектуру системи та її основні компоненти;

– розробити структуру бази даних та функціональні модулі;

– створити механізм адаптивного тестування та формування індивідуальної траєкторії навчання;

– реалізувати систему гейміфікації та користувацький інтерфейс;

– здійснити програмну реалізацію вебзастосунку;

– провести тестування розробленої системи та проаналізувати результати;

– підготувати інструкцію користувача.

Об’єктом дослідження є процес навчання іноземних мов із використанням цифрових технологій.

Предметом дослідження є методи та засоби розробки вебзастосунків для персоналізованого навчання іноземних мов.

Методи дослідження включають аналіз і узагальнення наукових джерел, системний аналіз, проєктування інформаційних систем, моделювання структури баз даних, об’єктно-орієнтоване програмування, а також функціональне й інтеграційне тестування програмного забезпечення.

Наукова новизна полягає у поєднанні механізмів адаптивного тестування, персоналізованого навчання та гейміфікації в межах єдиної вебсистеми, що дозволяє підвищити ефективність навчального процесу.

Практичне значення роботи полягає у створенні вебзастосунку, який може бути використаний для самостійного вивчення іноземних мов, у закладах освіти та системах дистанційного навчання.

Отримані результати можуть бути використані для подальшого розвитку адаптивних освітніх систем та удосконалення цифрових технологій навчання іноземних мов.

1 ТЕОРЕТИЧНІ ОСНОВИ СТВОРЕННЯ ВЕБЗАСТОСУНКІВ ДЛЯ ПЕРСОНАЛІЗОВАНОГО НАВЧАННЯ ІНОЗЕМНИХ МОВ

1.1 Особливості використання інформаційних технологій у вивченні іноземних мов

Інтеграція інформаційних технологій у процес навчання іноземних мов є одним із ключових напрямів модернізації сучасної освіти. Вона забезпечує якісно новий рівень організації навчального процесу, який характеризується інтерактивністю, індивідуалізацією та високою комунікативною спрямованістю. Використання цифрових інструментів сприяє підвищенню ефективності засвоєння мовного матеріалу та значно розширює можливості як здобувачів освіти, так і викладачів [1].

Однією з головних особливостей застосування інформаційних технологій у вивченні іноземних мов є індивідуалізація та диференціація навчання. Сучасні цифрові платформи дозволяють враховувати рівень підготовки кожного здобувача освіти, його темп навчання, сильні та слабкі сторони. Це забезпечує можливість побудови персоналізованої освітньої траєкторії, що є особливо важливим у мовній підготовці, де різниця у стартових знаннях студентів може бути суттєвою. У результаті навчальний процес стає більш гнучким і адаптивним до індивідуальних потреб користувача.

Важливим аспектом є також підвищення мотивації студентів до навчання. Використання мультимедійних технологій, інтерактивних вправ, мобільних застосунків та онлайн-платформ робить процес вивчення мови більш цікавим і наближеним до реальних життєвих ситуацій. Це сприяє формуванню стійкої внутрішньої мотивації та підвищенню залученості у навчальний процес [1]. Застосування елементів гри, візуалізації та інтерактивності дозволяє зменшити психологічний бар’єр у вивченні іноземної мови.

Інформаційні технології також сприяють розвитку продуктивних мовних навичок, таких як говоріння та письмо. Завдяки використанню спеціалізованих платформ здобувачі освіти мають можливість виконувати письмові завдання, отримувати автоматизований або викладацький зворотний зв'язок, а також тренувати усне мовлення за допомогою аудіо- та відеоматеріалів. Це дозволяє зробити процес навчання більш наочним, системним та практично орієнтованим [2].

Окрему роль відіграють цифрові інструменти для розширення словникового запасу та засвоєння граматичних структур. Такі мобільні застосунки, як Duolingo, Memrise або Babbel, використовують алгоритми повторення та тестування, що дозволяє ефективно закріплювати нову лексику та граматику. Подібні системи забезпечують регулярну практику та підтримують довготривале запам'ятовування навчального матеріалу.

Важливим компонентом сучасного навчання є розвиток навичок аудіювання. Для цього активно використовуються відеоплатформи, зокрема YouTube та TED Talks, які надають доступ до автентичного мовлення носіїв мови. Це дозволяє студентам адаптуватися до різних акцентів, темпу мовлення та реальних комунікативних ситуацій, що значно підвищує рівень розуміння усної мови [1; 3].

Інтерактивність та комунікативна спрямованість є одними з основних характеристик сучасних інформаційних технологій у мовній освіті. Вони забезпечують можливість організації віртуального спілкування, використання аудіо- та відеоконференцій, інтерактивних завдань і симуляцій мовленнєвих ситуацій. Це сприяє формуванню реальних комунікативних навичок та наближає навчальний процес до природного мовного середовища [3].

Крім того, інформаційні технології активізують пізнавальну діяльність студентів. Використання онлайн-ресурсів, електронних бібліотек, навчальних платформ і соціальних мереж у навчальних цілях сприяє розвитку самостійності, критичного мислення та дослідницьких навичок. Студенти отримують доступ до

великого обсягу автентичної інформації, що дозволяє розширювати їхній кругозір і мовну компетентність.

Не менш важливим є формування інформаційної культури здобувачів освіти. У процесі використання цифрових технологій студенти опановують навички роботи з інформацією, критичного її аналізу та ефективного використання у навчальних і професійних цілях. Це сприяє розвитку їхніх загальних компетентностей та підготовці до роботи в умовах цифрового суспільства [3].

Сучасний етап розвитку інформаційних технологій також характеризується активним впровадженням систем штучного інтелекту у навчальний процес. Такі системи можуть виконувати функції автоматичного аналізу помилок, генерації навчальних завдань та адаптації матеріалу до рівня користувача. Водночас важливо розглядати штучний інтелект як допоміжний інструмент, що доповнює традиційні методи навчання, а не замінює їх повністю [1].

Таким чином, створення сучасного інформаційно-освітнього середовища є необхідною умовою ефективного навчання іноземних мов. Воно забезпечує диференціацію навчання, підвищує його доступність та сприяє розвитку мовних компетентностей у здобувачів освіти.

Водночас слід зазначити, що впровадження інформаційних технологій супроводжується певними викликами. Серед них можна виділити необхідність належної цифрової компетентності викладачів і студентів, правильного добору навчальних ресурсів, а також забезпечення балансу між традиційними та цифровими методами навчання. Ефективне використання цифрових інструментів потребує методично обґрунтованого підходу та постійного оновлення педагогічних практик [4].

Отже, рівень володіння іноземною мовою значною мірою залежить від ефективності використання інформаційних технологій у навчальному процесі. Їх застосування дозволяє підвищити якість освіти, зробити навчання більш

індивідуалізованим, інтерактивним та результативним, що підтверджує їх ключову роль у сучасній системі мовної підготовки.

1.2 Аналіз сучасних платформ для вивчення іноземних мов та їх функціональних можливостей

Стрімкий розвиток інформаційних технологій та поширення мобільних пристроїв суттєво змінили підходи до вивчення іноземних мов. Сьогодні ринок EdTech-платформ для мовного навчання є одним із найдинамічніших сегментів галузі цифрової освіти. Станом на 2024–2025 роки провідними гравцями цього ринку є Duolingo, Babbel, Busuu, Memrise, Rosetta Stone, Mondly та Lingoda – кожна з яких реалізує власний підхід до організації навчального процесу, персоналізації та використання ігрових елементів [5].

Duolingo є найбільш масштабною мобільною платформою для вивчення мов у світі. Станом на кінець 2024 року платформа налічувала понад 100 мільйонів щомісячних активних користувачів та 8 мільйонів платних підписників [6]. Ключовою перевагою Duolingo є поєднання гейміфікації з адаптивним штучним інтелектом: система Birdbrain аналізує відповіді кожного користувача та коригує складність завдань у реальному часі, обробляючи близько 1 мільярда вправ на добу [7]. Платформа активно використовує елементи гейміфікації – серії (streaks), таблиці лідерів, нагороди за виконання завдань – що підвищує мотивацію та регулярність занять. У 2024 році в рамках конференції Duocon було представлено нові функції: відеодзвінки з ШІ-персонажем для розмовної практики та пригодницькі ігрові сценарії [8]. Водночас Duolingo отримує критику за поверховість системи інтервальних повторень: алгоритм повторення матеріалу не є достатньо оптимізованим для забезпечення довгострокового запам’ятовування на рівні, порівнянному зі спеціалізованими SRS-інструментами [9].

Babbel позиціонується як платформа для структурованого навчання, орієнтована насамперед на дорослих користувачів. Курси розроблено командою з понад 150 лінгвістів і методистів; зміст узгоджений із рівнями CEFR (A1–C1) [10]. Платформа зосереджена на формуванні практичних розмовних навичок через реалістичні сценарії та автентичний аудіоматеріал. Проте адаптивність у Babbel є обмеженою: персоналізація стосується переважно системи повторення помилок, тоді як сам навчальний маршрут залишається фіксованим для всіх користувачів [11]. Це робить платформу предсказуваною та зручною для тих, хто цінує чітку структуру, але недостатньо гнучкою для тих, чиї потреби суттєво відрізняються від середньостатистичного учня.

Busuu виділяється серед конкурентів завдяки унікальному поєднанню структурованого CEFR-узгодженого курсу та соціальної мережі для обміну зворотним зв’язком із носіями мови [12]. Платформа надає можливість отримати виправлення письмових і усних вправ від представників спільноти, що є цінним для розвитку продуктивних навичок. Busuu включає тест на визначення рівня знань і персоналізований план навчання, а навчальний контент розбитий на короткі уроки з відеовставками та контрольними тестами [13]. Недоліком є те, що зворотний зв’язок від користувачів не є миттєвим, а якість відповідей може суттєво варіюватися залежно від активності спільноти.

Memrise спеціалізується на запам’ятовуванні лексики за допомогою відеофрагментів із реальними носіями мови та методу інтервальних повторень (Spaced Repetition System, SRS) [14]. Дослідження засвідчують, що SRS є надзвичайно ефективним методом для довгострокового запам’ятовування лексичних одиниць: застосування такого підходу в мобільних додатках сприяє підвищенню рівня засвоєння словникового запасу порівняно з традиційними методами [9]. Однак Memrise не забезпечує повноцінного охоплення граматики та комплексної мовної підготовки, а якість курсів є нерівномірною. Платформа краще підходить як доповнення до основного навчання, аніж як самостійний інструмент.

Rosetta Stone – одна з найстаріших і найвідоміших платформ для вивчення мов, яка ґрунтується на методі занурення (Dynamic Immersion): учень засвоює мову через контекстуальні зображення та аудіо без перекладу рідною мовою, імітуючи природний процес оволодіння мовою [15]. Платформа оснащена технологією розпізнавання мовлення TruAccent, яка забезпечує миттєвий зворотний зв'язок щодо вимови. Проте відсутність граматичних пояснень та перекладу робить платформу менш ефективною для дорослих учнів, яким потрібне усвідомлення мовних структур. Інтерфейс і методологія значно поступаються новішим конкурентам із точки зору застосування ШІ та адаптивності [16].

Mondly, придбана компанією Pearson у 2022 році, пропонує короткі щоденні уроки тривалістю 5–15 хвилин і охоплює 41 мову [17]. Відмінністю Mondly є використання AR/VR-технологій для занурення в симульоване мовне середовище, а також чат-бот для діалогової практики. Платформа містить тест на визначення рівня знань та галузеві пакети лексики. Серед слабких сторін – відносно поверхова адаптивність і менш розвинена система аналітики навчального прогресу порівняно з Duolingo.

Lingoda відрізняється від усіх перелічених платформ: вона функціонує як повноцінна онлайн-школа з живими заняттями під керівництвом викладачів – носіїв мови або сертифікованих спеціалістів [18]. Заняття проводяться у малих групах (зазвичай 3–4 студенти) або в індивідуальному форматі та будуються за CEFR-узгодженою програмою від A1 до C1. Lingoda забезпечує найвищий рівень персоналізованого зворотного зв'язку серед розглянутих платформ, однак є найдорожчою опцією і не містить ігрових або адаптивних елементів у класичному розумінні.

Ключовою технологічною тенденцією для всіх сучасних платформ є активне впровадження ШІ для персоналізації навчального досвіду. Адаптивні навчальні технології пристосовуються до студента в реальному часі, забезпечуючи аналітику навчального процесу як для учня, так і для платформи [19]. Гейміфікація – використання ігрових механік, балів, рівнів і

нагород – є спільною рисою більшості платформ і, за даними досліджень, здатна підвищити залученість користувачів до 60% [1620 Інтервальне повторення (SRS), реалізоване в Memrise, Anki та частково Duolingo, є науково обґрунтованим методом, що показав високу ефективність у довгострокових дослідженнях засвоєння лексики [9].

Попри широкий спектр наявних рішень, аналіз платформ виявляє системні недоліки. По-перше, більшість платформ пропонує або глибоку адаптивність (Duolingo, Memrise), або якісний зміст із чіткою методологією (Babbel, Lingoda), але не поєднує обидва підходи в одному продукті. По-друге, адаптивне тестування в таких системах, як Duolingo, орієнтоване переважно на управління темпом навчання, а не на діагностику конкретних прогалин у знаннях учня. По-третє, ігрові елементи нерідко слугують інструментом утримання користувачів (retention), а не безпосередньо дидактичним засобом. По-четверте, адаптивне навчання застосовується недостатньо довго, щоб можна було отримати надійні результати лонгітюдних досліджень, а самé поняття персоналізації тлумачиться по-різному різними платформами [21]. Зазначені прогалини окреслюють простір для розробки вебзастосунку, який поєднає глибоке адаптивне тестування, науково обґрунтовані SRS-алгоритми та змістовно інтегровану гейміфікацію в єдину когерентну навчальну систему.

Порівняльний аналіз функціональних можливостей платформ наведено в таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз функціональних можливостей платформ для вивчення іноземних мов

Платформа	**Адаптивність**	**Гейміфікація**	**SRS**	**CEFR**	**Соц. спільнота**	**ШІ-функції**	**Модель доступу**
Duolingo	Висока (Birdbrain)	Висока	Частково	Ні	Ні	Висока (ШІ-діалоги)	Freemium
Babbel	Низька	Середня	Ні	Так	Ні	Низька	Платна
Busuu	Середня	Середня	Частково	Так	Так	Середня	Freemium

Memrise	Середня (SRS)	Середня	Висока	Ні	Ні	Низька	Freemium
Rosetta Stone	Низька	Низька	Ні	Ні	Ні	Низька	Платна
Mondly	Середня	Середня	Часткова	Ні	Ні	Середня (AR/VR)	Freemium
Lingoda	Висока (вчитель)	Низька	Ні	Так	Так	Низька	Платна

У таблиці 1.1 наведено порівняльний аналіз основних сучасних платформ для вивчення іноземних мов за ключовими функціональними характеристиками. Зокрема, розглянуто рівень адаптивності систем, наявність гейміфікації, використання інтервального повторення (SRS), відповідність стандарту CEFR, наявність соціальної взаємодії, застосування функцій штучного інтелекту та модель доступу.

Аналіз показує, що найбільш технологічно розвиненою з точки зору адаптивності та ШІ є Duolingo, тоді як Memrise демонструє найкращі результати у використанні SRS. Платформи Babbel та Lingoda орієнтовані на структуроване навчання за CEFR, але мають обмежену або відсутню гейміфікацію. Busuu поєднує соціальне навчання з помірною адаптивністю, а Mondly та Rosetta Stone використовують більш традиційні підходи з частковим впровадженням інноваційних технологій.

Загалом таблиця демонструє, що жодна з платформ не забезпечує одночасно високий рівень адаптивності, гейміфікації та ефективного SRS в єдиній системі, що підкреслює актуальність розробки нового комплексного рішення.

Отже, аналіз сучасних платформ для вивчення іноземних мов показав високий рівень розвитку EdTech-рішень та активне використання штучного інтелекту, адаптивного навчання і гейміфікації. Водночас більшість систем реалізують ці підходи частково і не забезпечують їх повної інтеграції в єдину архітектуру. Таким чином, існуючі обмеження підтверджують актуальність

розробки вебзастосунку, який поєднуватиме адаптивне тестування, персоналізоване навчання та гейміфікацію в одному рішенні.

1.3 Визначення вимог до вебзастосунку персоналізованого навчання іноземних мов

На основі аналізу предметної області, сучасних підходів до цифрового навчання та досвіду провідних EdTech-платформ сформовано вимоги до вебзастосунку LinguaQuest. Вони визначають логіку роботи системи, її функціональні можливості, якісні характеристики та структуру взаємодії користувачів із платформою.

Функціональні вимоги (табл. 1.2) описують основні можливості системи, тобто те, що саме повинен уміти робити вебзастосунок для забезпечення навчального процесу. Вони охоплюють усі етапи взаємодії користувача з платформою: від реєстрації до аналізу результатів навчання.

Таблиця 1.2 – Функціональні вимоги вебзастосунку LinguaQuest

Код	Вимога	Роль
ФВ-1	Реєстрація та автентифікація користувача (email + пароль)	Усі
ФВ-2	Розмежування доступу за трьома ролями	Усі
ФВ-3	Проходження адаптивного тесту зі зміною складності	Учень
ФВ-4	Миттєвий зворотний зв’язок щодо правильності відповіді	Учень
ФВ-5	Накопичення балів досвіду (XP) та підвищення рівнів	Учень
ФВ-6	Перегляд власного прогресу та таблиці лідерів	Учень
ФВ-7	Формування персоналізованих рекомендацій	Учень
ФВ-8	Створення та редагування банку тестових питань	Вчитель
ФВ-9	Керування обліковими записами та ролями	Адмін
ФВ-10	Перегляд статистики платформи	Адмін

У таблиці 1.2 представлено ключові функціональні можливості системи, які відображають повний цикл роботи вебзастосунку.

Насамперед, система забезпечує базові функції доступу – реєстрацію та автентифікацію користувачів (ФВ-1), що є необхідною умовою для

персоналізованого навчання та збереження прогресу. Розмежування доступу за ролями (ФВ-2) дозволяє розділити функціонал між учнем, вчителем та адміністратором, забезпечуючи безпеку та логічну структуру системи.

Основний навчальний функціонал зосереджений на ролі учня. Зокрема, адаптивне тестування (ФВ-3) дозволяє автоматично змінювати складність завдань залежно від відповідей користувача, що забезпечує індивідуалізацію навчального процесу. Миттєвий зворотний зв’язок (ФВ-4) сприяє швидкому закріпленню знань та виправленню помилок.

Гейміфікаційний компонент реалізовано через систему балів досвіду (XP), рівнів та досягнень (ФВ-5), що підвищує мотивацію користувача. Додатково користувач може переглядати власний прогрес і рейтинги (ФВ-6), що стимулює конкурентність та регулярність навчання.

Функція формування персоналізованих рекомендацій (ФВ-7) забезпечує адаптацію навчального контенту відповідно до результатів користувача, що є ключовим елементом концепції персоналізованого навчання.

Для викладачів передбачено можливість створення та редагування банку тестових завдань (ФВ-8), що дозволяє формувати навчальний контент і контролювати його якість. Адміністратор системи здійснює керування користувачами та ролями (ФВ-9), а також переглядає загальну статистику використання платформи (ФВ-10), що необхідно для моніторингу ефективності системи.

Нефункціональні вимоги (табл. 1.3) визначають якісні характеристики системи, які впливають на зручність, швидкість, безпеку та стабільність роботи вебзастосунку.

Таблиця 1.3 – Нефункціональні вимоги системи LinguaQuest

Код	Категорія	Вимога
НВ-1	Зручність	Інтуїтивний адаптивний інтерфейс (mobile-first)
НВ-2	Безпека	Хешування паролів, захищені сесії (HTTP-only cookie)
НВ-3	Продуктивність	Швидкий відгук завдяки серверному рендерингу
НВ-4	Надійність	Коректне збереження результатів у базі даних
НВ-5	Масштабованість	Модульна архітектура для додавання нових мов і тем
НВ-6	Сумісність	Коректна робота в сучасних браузерах

У таблиці 1.3 наведено нефункціональні вимоги, які визначають якість роботи системи, а не її конкретні функції.

Зручність використання (НВ-1) передбачає адаптивний інтерфейс, оптимізований для мобільних пристроїв, що є критично важливим для сучасних освітніх платформ. Безпека (НВ-2) забезпечується шляхом використання хешування паролів та захищених сесій, що запобігає несанкціонованому доступу до даних користувачів.

Продуктивність системи (НВ-3) визначається швидкістю відгуку інтерфейсу та обробки запитів, що реалізується через серверний рендеринг. Надійність (НВ-4) гарантує коректне збереження навчальних результатів і прогресу користувачів без втрати даних.

Масштабованість (НВ-5) передбачає можливість розширення системи шляхом додавання нових мов, курсів та типів завдань без суттєвої зміни архітектури. Сумісність (НВ-6) забезпечує коректну роботу вебзастосунку у всіх сучасних браузерах та на різних пристроях.

У межах підрозділу визначено основні вимоги до вебзастосунку персоналізованого навчання іноземних мов LinguaQuest, які охоплюють функціональні можливості системи, її якісні характеристики та структуру ролей користувачів. Функціональні вимоги відображають ключові сценарії роботи платформи, зокрема адаптивне тестування, систему гейміфікації, персоналізовані рекомендації та інструменти управління навчальним контентом. Нефункціональні вимоги визначають умови ефективної роботи системи, включаючи безпеку, продуктивність, масштабованість та зручність використання.

2 ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ ДЛЯ ПЕРСОНАЛІЗОВАНОГО НАВЧАННЯ ІНОЗЕМНИХ МОВ

2.1 Аналіз предметної області та проєктування архітектури системи

Наступним етапом аналізу предметної області є визначення користувачів системи та їхніх функціональних можливостей. Це дозволяє сформувати логіку взаємодії між користувачем і платформою та визначити межі доступу до функцій.

У системі LinguaQuest виділено три основні ролі (рис. 2.1):

Учень (Student) – основний користувач системи, який проходить адаптивні тести, отримує зворотний зв’язок, накопичує бали досвіду (XP) та переглядає власний прогрес.

Вчитель (Teacher) – відповідає за створення, редагування та наповнення банку тестових завдань, а також формування навчального контенту.

Адміністратор (Admin) – здійснює управління користувачами, ролями та загальним функціонуванням платформи, а також переглядає аналітичну статистику.

Таким чином, система реалізує класичну рольову модель доступу, яка забезпечує безпеку, структурованість і розмежування відповідальності між користувачами.

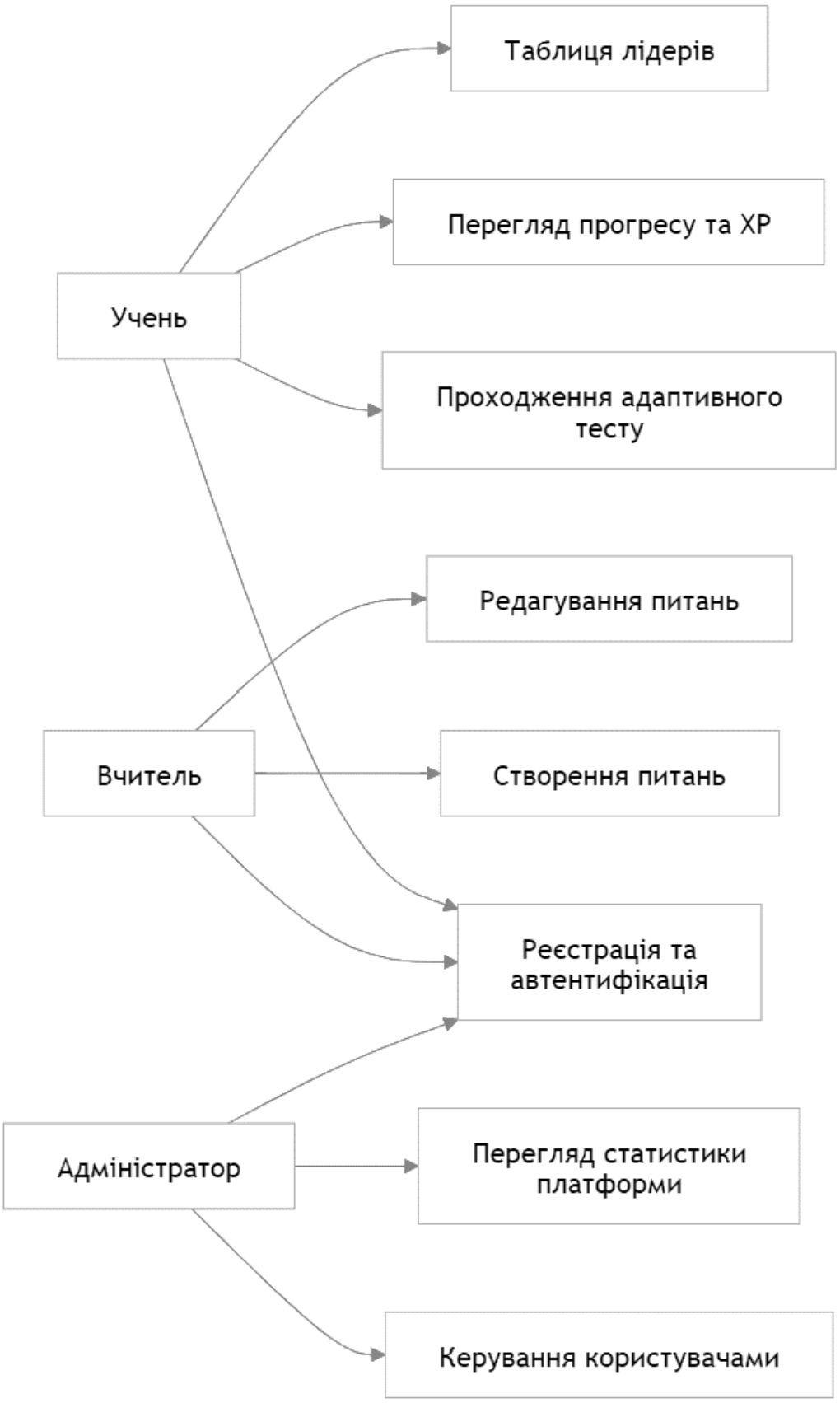


Рисунок 2.1 – Use Case діаграма взаємодії користувачів із системою LinguaQuest

Діаграма відображає основні сценарії використання системи та взаємодію різних ролей із функціональними можливостями платформи.

Учень взаємодіє з функціями навчання: проходження адаптивних тестів, перегляд прогресу, накопичення XP та перегляд рейтингу.

Вчитель працює з навчальним контентом, створюючи та редагуючи тестові завдання.

Адміністратор здійснює керування користувачами та аналіз статистики системи.

Після аналізу предметної області та визначення ролей користувачів виконується проєктування архітектури вебзастосунку. Основною метою цього етапу є визначення логічної структури системи, розподілу відповідальності між її компонентами та способу взаємодії між рівнями.

Вебзастосунок LinguaQuest побудовано на основі багаторівневої архітектури, яка включає три основні рівні (рис. 2.2).

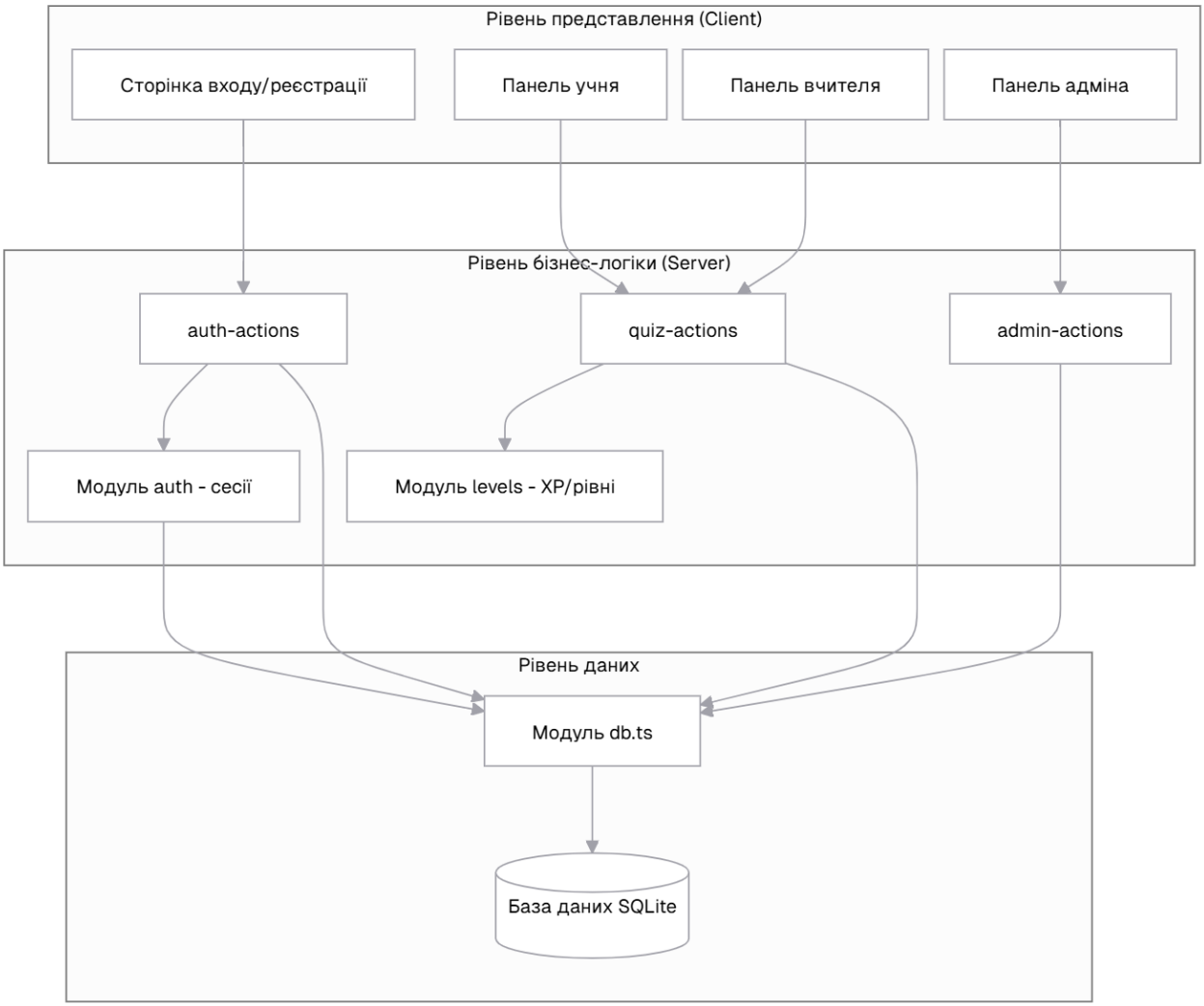


Рисунок 2.2 – Архітектура вебзастосунку

Подана діаграма відображає трирівневу архітектуру вебзастосунку, яка складається з рівня представлення, рівня бізнес-логіки та рівня даних.

Рівень представлення (Client) включає сторінки входу та реєстрації, панелі учня, вчителя та адміністратора. На цьому рівні користувач взаємодіє із системою: проходить тести, переглядає прогрес, керує завданнями або адмініструє платформу залежно від ролі.

Рівень бізнес-логіки (Server) містить основні серверні модулі системи: auth-actions для автентифікації та керування сесіями, quiz-actions для обробки тестів і результатів, admin-actions для адміністративних функцій, а також модуль levels, що відповідає за систему XP та рівнів. Цей рівень реалізує основну логіку персоналізованого навчання.

Рівень даних представлений базою даних SQLite та модулем db.ts, які забезпечують зберігання та доступ до інформації про користувачів, тести, результати та прогрес навчання.

Отже, проаналізовано предметну область персоналізованого навчання іноземних мов та визначено основні вимоги до вебзастосунку LinguaQuest. Спроєктовано трирівневу архітектуру системи, що включає клієнтський рівень, серверну логіку та рівень даних. Це забезпечує чітку структуру системи, її масштабованість і основу для подальшої реалізації функціоналу.

2.2 Проєктування бази даних та функціональних модулів вебзастосунку

Концептуальна модель бази даних визначає основні сутності системи та зв’язки між ними. У межах проєктованого вебзастосунку виділено чотири ключові сутності: користувачі, питання, сесії та результати тестування (табл. 2.1).

Таблиця 2.1 – Основні сутності бази даних системи LinguaQuest

Сутність	Опис	Ключові атрибути
users	Користувачі всіх ролей	id, name, email, role, xp
questions	Банк тестових питань	id, text, options, correct_index, difficulty, topic
sessions	Активні сесії автентифікації	id, user_id, created_at
test_results	Результати пройдених тестів	id, user_id, total, correct, xp_earned, final_level

У таблиці наведено логічний опис структури даних, що використовується в системі. Сутність users містить інформацію про всіх користувачів платформи (учнів, вчителів та адміністраторів), включаючи їхні облікові дані, роль та накопичені бали досвіду (XP). Сутність questions відповідає за банк тестових завдань і містить текст питання, варіанти відповідей, правильну відповідь, рівень складності та тему. Таблиця sessions використовується для керування активними сесіями автентифікації користувачів, що забезпечує безпечний доступ до

системи. Сутність test_results фіксує результати проходження тестів, включаючи кількість правильних відповідей, загальний бал, отриманий XP та фінальний рівень користувача.

Наступним етапом проєктування є формування логічної моделі даних, яка відображає зв’язки між сутностями системи. Для цього побудовано ER-діаграму, що демонструє взаємодію основних таблиць бази даних (рис. 2.3).

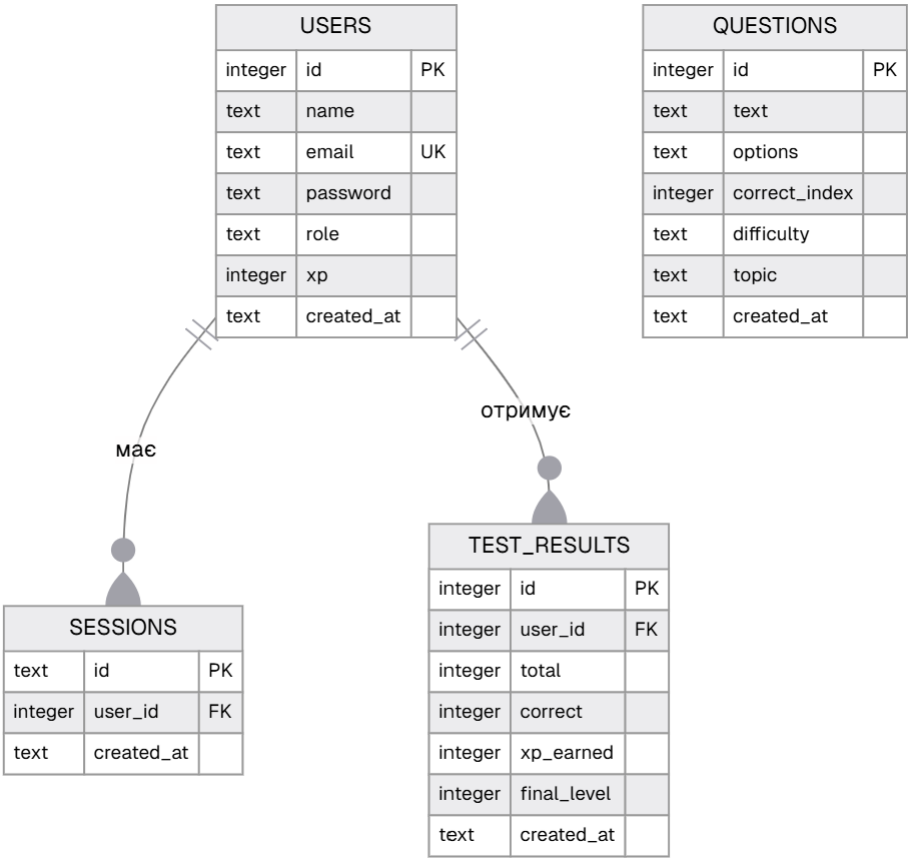


Рисунок 2.3 – ER-діаграма бази даних

Діаграма ілюструє зв’язки між користувачами та їхніми результатами тестування, а також взаємодію з банком питань і сесіями автентифікації. Зокрема, один користувач може мати багато результатів тестів, а кожен тестовий запис пов’язаний із конкретним користувачем через зовнішній ключ user_id. Така структура забезпечує цілісність даних і дозволяє ефективно відстежувати навчальний прогрес.

Детальне проєктування бази даних включає опис структури основних таблиць, їхніх полів та обмежень.

Таблиця users (табл. 2.2) призначена для зберігання даних про користувачів системи. Вона містить як базову інформацію (ім’я, email, пароль), так і навчальні показники, зокрема рівень досвіду (XP). Поле role визначає права доступу користувача та впливає на доступні функціональні можливості системи. Унікальність email забезпечує коректну ідентифікацію користувача, а хешування пароля підвищує рівень безпеки.

Таблиця 2.2 – Структура таблиці users

Поле	Тип	Обмеження	Опис
id	INTEGER	PK, AUTOINCREMENT	Унікальний ідентифікатор
name	TEXT	NOT NULL	Ім’я користувача
email	TEXT	NOT NULL, UNIQUE	Електронна пошта (логін)
password	TEXT	NOT NULL	Хеш пароля
role	TEXT	CHECK (student/teacher/admin)	Роль користувача
xp	INTEGER	DEFAULT 0	Бали досвіду
created_at	TEXT	DEFAULT datetime(‘now’)	Дата реєстрації

Аналогічно, таблиця questions забезпечує зберігання навчального контенту, що використовується під час тестування.

У таблиці questions (табл. 2.3) зберігається весь банк тестових завдань, який є основою навчального процесу. Кожне питання має текстову частину, набір варіантів відповідей у форматі JSON, індекс правильної відповіді, рівень складності та тематичну приналежність. Така структура дозволяє реалізувати адаптивне тестування та фільтрацію завдань за рівнем знань користувача.

Таблиця 2.3 – Структура таблиці questions

Поле	Тип	Обмеження	Опис
id	INTEGER	PK, AUTOINCREMENT	Ідентифікатор питання
text	TEXT	NOT NULL	Текст питання
options	TEXT	NOT NULL	Варіанти відповідей
correct_index	INTEGER	NOT NULL	Правильна відповідь
difficulty	TEXT	CHECK (beginner/intermediate/advanced)	Рівень складності
topic	TEXT	NOT NULL	Тема питання

На основі аналізу предметної області система була декомпонована на окремі функціональні модулі, кожен з яких відповідає за певний набір задач (рис. 2.4).

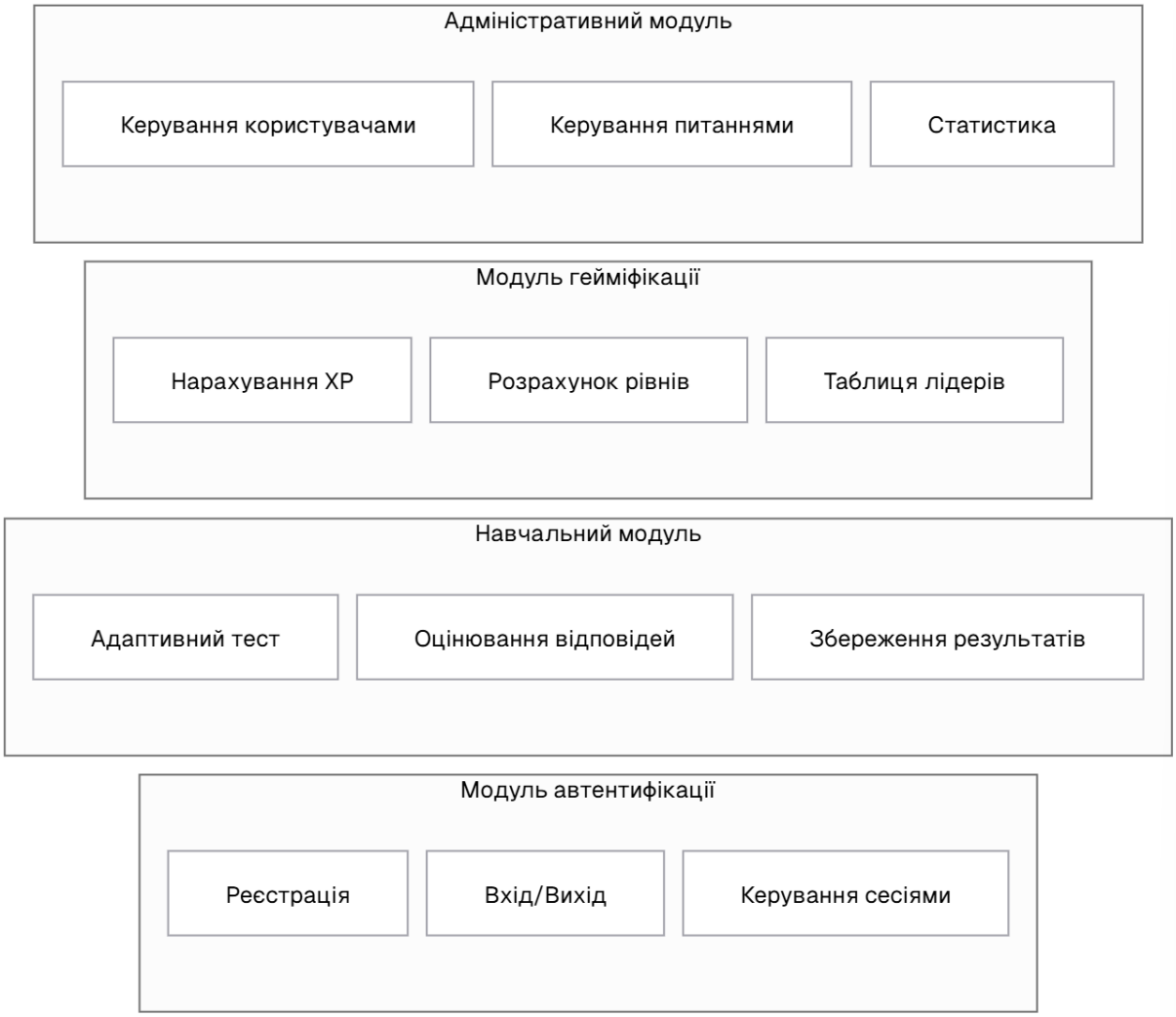


Рисунок 2.4 – Функціональні модулі

Модуль автентифікації відповідає за реєстрацію, вхід у систему та керування сесіями користувачів. Навчальний модуль реалізує адаптивне тестування, перевірку відповідей і збереження результатів. Модуль гейміфікації забезпечує нарахування XP, розрахунок рівнів та формування рейтингової таблиці. Адміністративний модуль призначений для вчителів та адміністраторів

і дозволяє керувати користувачами, тестовими питаннями та переглядати статистику платформи.

Таким чином, спроєктована база даних та виділені функціональні модулі формують цілісну основу вебзастосунку LinguaQuest. Вони забезпечують узгоджену взаємодію між збереженням даних, бізнес-логікою та користувацьким інтерфейсом, що є необхідним для подальшої реалізації адаптивного навчального функціоналу.

2.3 Проєктування механізму адаптивного тестування та формування індивідуальної траєкторії навчання

Проєктований механізм адаптивного тестування у вебзастосунку базується на принципах динамічної зміни складності навчальних завдань залежно від результатів відповідей користувача. Основною метою такого підходу є забезпечення персоналізації навчального процесу та підвищення ефективності засвоєння матеріалу за рахунок автоматичного підлаштування рівня складності під реальні знання учня.

В основі реалізованого рішення лежить спрощена модель адаптивного тестування, яка передбачає поділ усіх тестових завдань на три рівні складності: початковий, середній та просунутий. Кожен рівень має відповідний код та визначену кількість балів (XP), що нараховуються за правильну відповідь.

Тестування завжди розпочинається з питання початкового рівня, що дозволяє оцінити базову підготовку користувача без завищеного стартового навантаження. Подальша логіка адаптації ґрунтується на простому правилі корекції складності: у разі правильної відповіді складність наступного питання підвищується на один рівень, у разі неправильної – знижується. При цьому передбачено обмеження на межах шкали складності, де рівень не може виходити за межі beginner та advanced.

Такий підхід забезпечує поступове ускладнення або спрощення завдань відповідно до реального рівня підготовки учня, що є основою адаптивного навчання.

Алгоритм функціонування адаптивного тестування представлено у вигляді блок-схеми (рис. 2.5), яка відображає повний цикл проходження тесту від ініціалізації до збереження результатів.

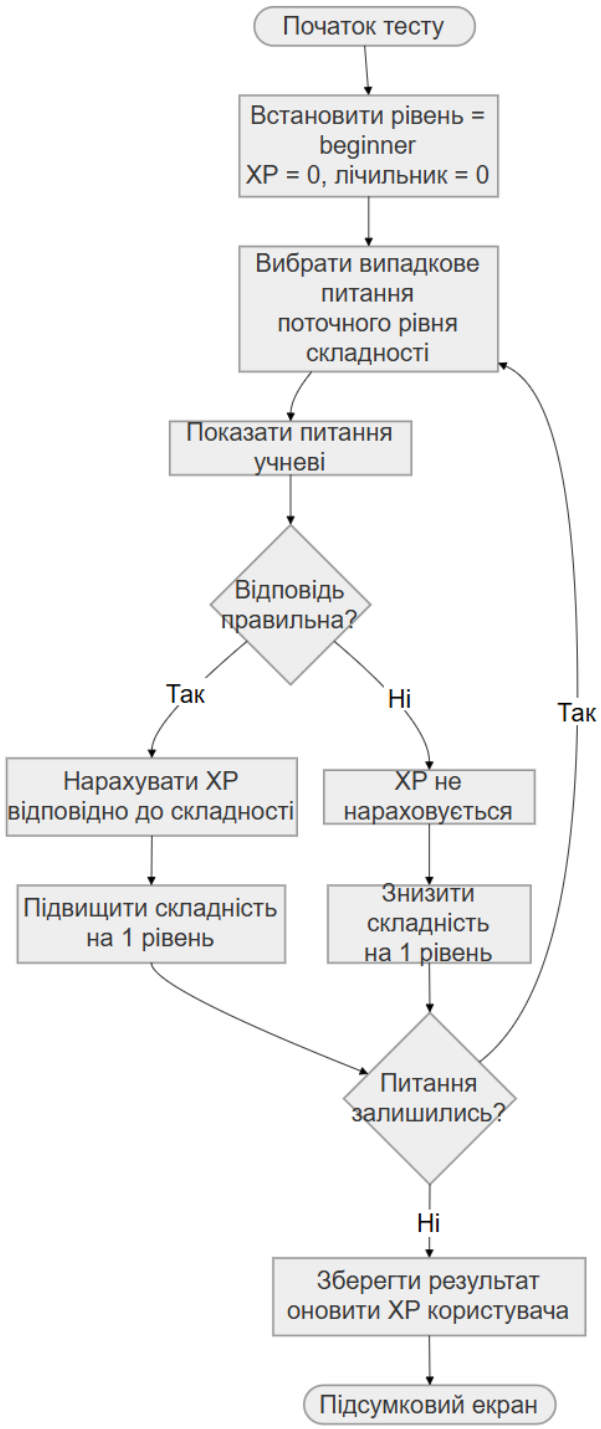


Рисунок 2.5 – Алгоритм роботи адаптивного тесту

Подана діаграма демонструє послідовність виконання тестування. На початковому етапі система ініціалізує параметри тесту, встановлюючи рівень складності на значення beginner, а також обнуляючи показники XP та лічильник запитань.

Далі система випадковим чином обирає питання відповідного рівня складності та відображає його користувачу. Після отримання відповіді виконується перевірка її правильності. У випадку правильної відповіді користувачу нараховуються XP відповідно до складності питання, після чого рівень складності підвищується. У разі помилки XP не нараховується, а рівень складності знижується.

Процес повторюється доти, доки не будуть вичерпані всі питання тесту. Після завершення тестування система зберігає результат, оновлює загальний XP користувача та відображає підсумковий екран.

Таким чином, алгоритм забезпечує динамічну зміну складності в реальному часі та формує адаптивну траєкторію проходження тесту.

Однією з ключових особливостей розробленого механізму є формування індивідуальної траєкторії навчання, яка залежить від поведінки користувача під час тестування. Завдяки адаптивній логіці кожен учень проходить унікальну послідовність завдань, що виключає однаковість тестових сценаріїв.

Приклад формування різних траєкторій наведено в таблиці 2.4.

Таблиця 2.4 – Приклад індивідуальних траєкторій навчання

Крок	**Учень А (успішні відповіді)**	**Учень Б (помилки)**
1	beginner = +	beginner = -
2	intermediate = +	beginner = +
3	advanced = +	intermediate = -
4	advanced = -	beginner = +
Підсумок	Підвищена складність, високий XP	Закріплення базового рівня

Як видно з наведеного прикладу, учні з різним рівнем підготовки автоматично отримують різний набір завдань. Сильніші користувачі швидше

переходять до складних питань, тоді як користувачі, які допускають помилки, отримують більше завдань базового рівня для закріплення знань.

Такий підхід забезпечує реалізацію принципу персоналізації навчання та підвищує ефективність засвоєння матеріалу.

Логіку взаємодії користувача із системою під час проходження тесту відображено на діаграмі послідовності (рис. 2.6).

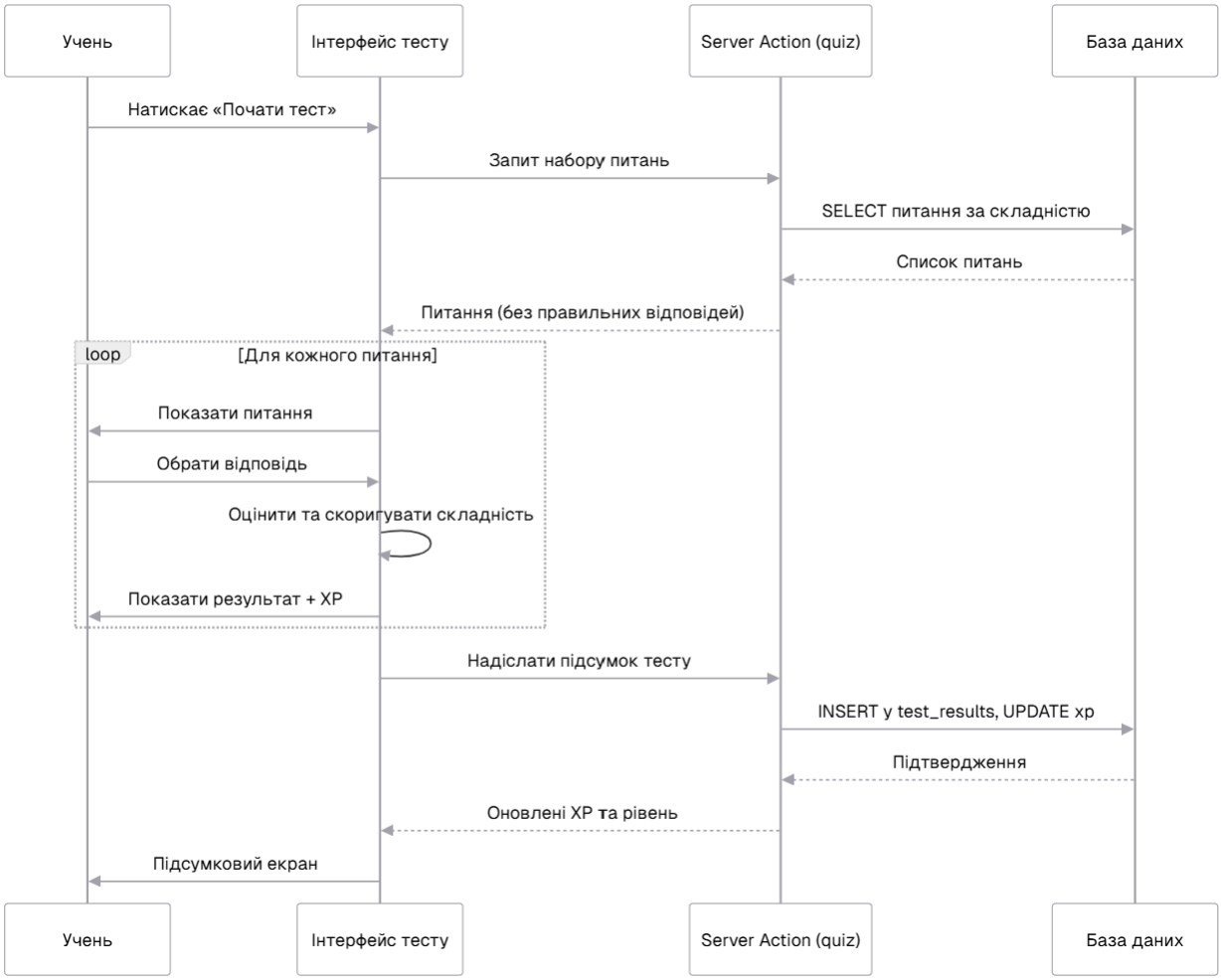


Рисунок 2.6 – Діаграма послідовності проходження адаптивного тесту

Подана діаграма ілюструє взаємодію між користувачем, інтерфейсом тесту, серверною логікою та базою даних. Процес починається з ініціації тесту користувачем через інтерфейс.

Далі серверний модуль здійснює запит до бази даних для отримання набору питань відповідно до поточного рівня складності. Отримані питання передаються до інтерфейсу користувача, де відбувається їх відображення.

Після вибору відповіді користувачем серверна логіка виконує перевірку правильності та оновлює стан тесту. У разі правильної відповіді нараховуються XP, а також коригується рівень складності. Результати тестування зберігаються у таблиці результатів, після чого оновлюється загальний прогрес користувача.

Завершальним етапом є формування підсумкового екрану, на якому відображаються набрані бали, рівень та загальний результат тестування.

Отже, розроблений механізм адаптивного тестування забезпечує динамічну зміну складності завдань та формування індивідуальної траєкторії навчання для кожного користувача. Використання простої, але ефективної логіки корекції складності дозволяє реалізувати персоналізований підхід без значного ускладнення архітектури системи, що підвищує її ефективність та практичну доцільність.

2.4 Проєктування системи гейміфікації та користувацького інтерфейсу

У процесі проєктування вебзастосунку особлива увага приділяється не лише навчальній логіці, а й мотиваційним та візуальним компонентам системи. Саме тому було розроблено підсистему гейміфікації та користувацького інтерфейсу, які забезпечують залученість користувачів, підвищують рівень їхньої активності та формують позитивний навчальний досвід.

Система гейміфікації побудована на двох базових елементах: балах досвіду (XP) та рівнях користувача. Нарахування XP здійснюється за кожну правильну відповідь і залежить від складності питання, що дозволяє об’єктивно оцінювати прогрес учня.

Розрахунок рівня користувача виконується за формулою:

$$L = \left[\frac{XP}{100}\right] + 1$$

де L – поточний рівень користувача, а XP – загальна кількість набраних балів досвіду. Така модель передбачає, що кожні 100 балів досвіду відповідають переходу на новий рівень.

Додатково визначається відсоток прогресу до наступного рівня:

$$P = \frac{XP mod 100}{100} * 100\%$$

Шкала рівнів користувача представлена в таблиці 2.5.

Таблиця 2.5 – Шкала рівнів користувача

Рівень	**Діапазон XP**	**Умовна назва**
1	0–99	Початківець
2	100–199	Учень
3	200–299	Впевнений
4	300–399	Досвідчений
5+	400+	Майстер

Таким чином, система рівнів створює довгострокову мотивацію до навчання, оскільки користувач постійно бачить власний прогрес і має чітку ціль для подальшого розвитку.

Логіка роботи системи гейміфікації представлена у вигляді діаграми (рис. 2.7), яка відображає повний цикл оновлення прогресу користувача після кожної відповіді.

Подана діаграма демонструє, що після отримання правильної відповіді система ініціює нарахування XP відповідно до рівня складності питання. Далі відбувається оновлення сумарного XP у таблиці користувачів.

Наступним етапом виконується перерахунок рівня користувача за наведеною формулою та визначення прогресу до наступного рівня. Отримані дані використовуються для оновлення інтерфейсу користувача, зокрема прогрес-бару та таблиці лідерів.

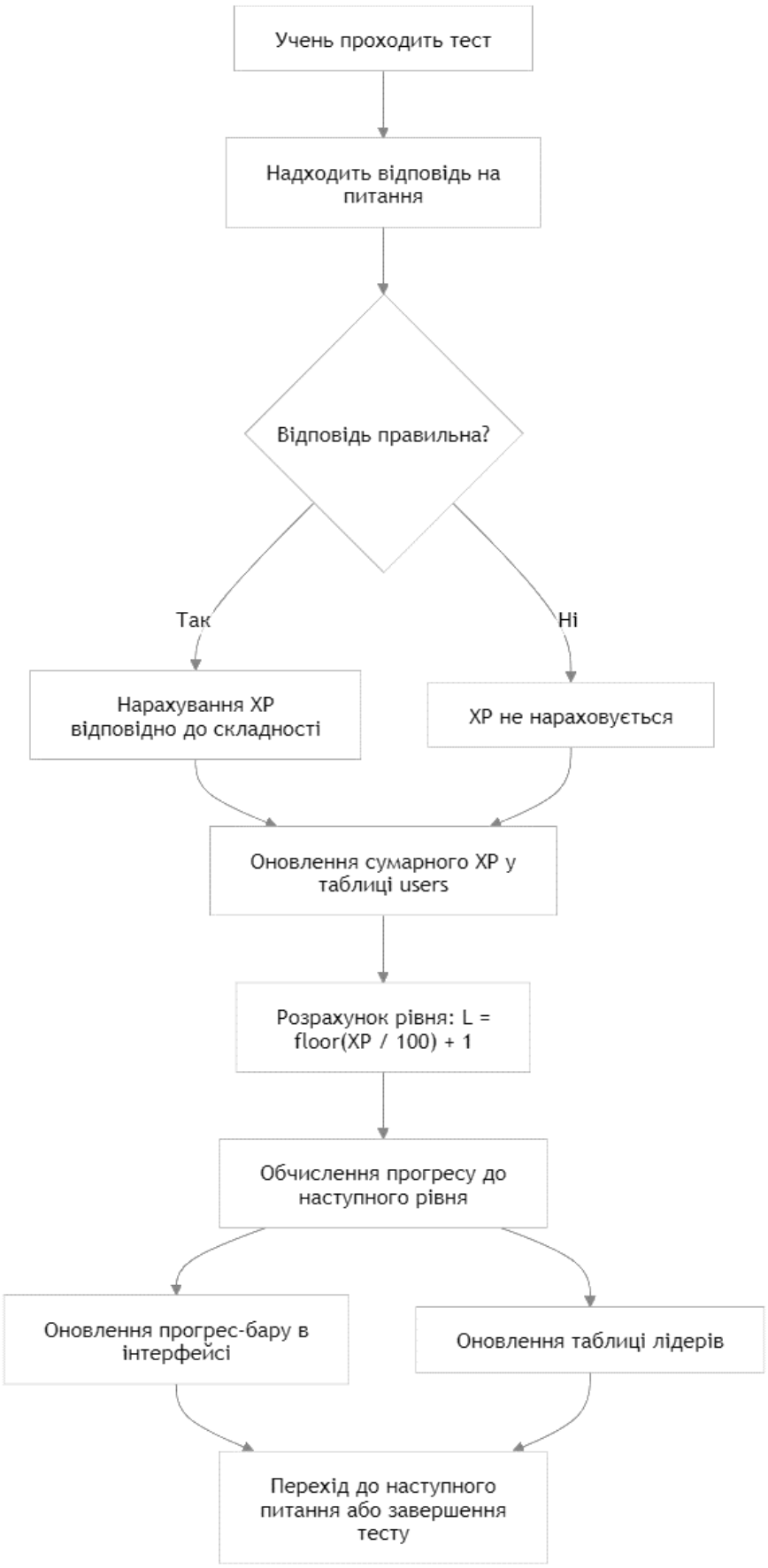


Рисунок 2.7 – Логіка нарахування XP та оновлення рівня користувача

Окремим елементом системи є таблиця лідерів, яка формує рейтинг користувачів на основі загальної кількості XP. Це додає елемент змагання між учнями та підвищує їхню мотивацію до регулярного навчання.

Користувацький інтерфейс вебзастосунку розроблено з урахуванням принципів зручності, мінімалізму та адаптивності. Основним підходом обрано mobile-first дизайн, що дозволяє забезпечити коректне відображення системи як на мобільних пристроях, так і на великих екранах.

Візуальна частина інтерфейсу побудована на синій кольоровій палітрі, яка психологічно асоціюється зі стабільністю, довірою та концентрацією уваги – ключовими факторами для навчального середовища.

Кольорова система представлена в таблиці 2.6.

Таблиця 2.6 – Кольорова схема інтерфейсу

Елемент	**Призначення**	**Тон**
Primary	Основні кнопки та акценти	Насичений синій
Secondary	Допоміжні елементи	Світло-синій
Background	Фон сторінок	Світлий із синім відтінком
Muted	Вторинний текст	Сіро-синій
Foreground	Основний текст	Темно-синій

Такий підхід забезпечує візуальну узгодженість інтерфейсу та знижує когнітивне навантаження на користувача під час роботи з системою.

Навігаційна структура системи (рис. 2.8) побудована з урахуванням ролей користувачів, які визначаються після проходження автентифікації. В залежності від ролі система формує відповідний набір доступних сторінок та функціональних можливостей.

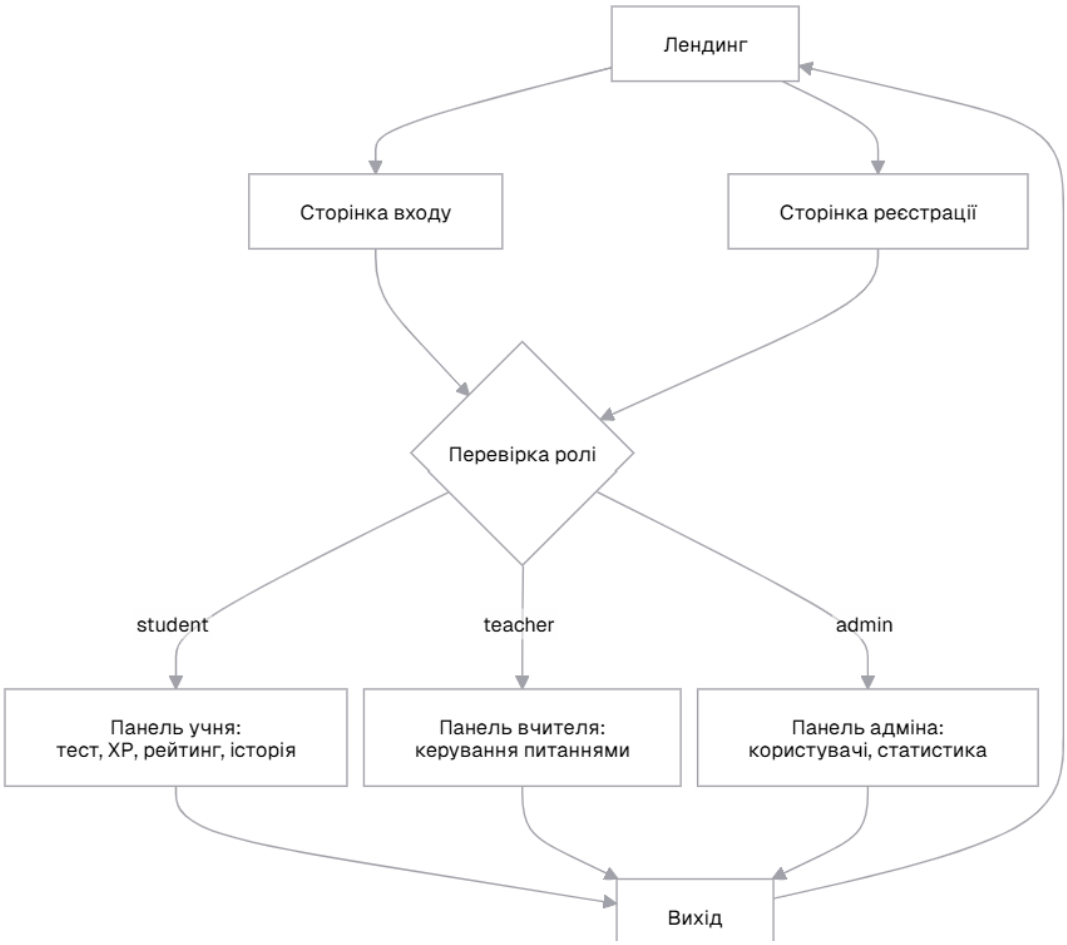


Рисунок 2.8 – Структура навігації вебзастосунку

Діаграма відображає логіку переходів між основними сторінками системи. Початковим етапом є лендинг-сторінка, після чого користувач переходить до сторінки входу або реєстрації.

Після автентифікації виконується перевірка ролі користувача, на основі якої система перенаправляє його до відповідної панелі: учня, вчителя або адміністратора. Учень отримує доступ до проходження тестів, перегляду XP та рейтингу, вчитель – до керування питаннями, а адміністратор – до керування користувачами та системною статистикою.

Завершальним етапом є можливість виходу із системи та завершення сесії.

Також інтерфейс вебзастосунку реалізовано за компонентним підходом (табл. 2.7), що забезпечує повторне використання елементів та спрощує подальшу підтримку системи.

Таблиця 2.7 – Основні UI-компоненти системи

Компонент	**Призначення**	**Доступ**
dashboard-header	Верхня панель навігації та вихід	Усі ролі
student-dashboard	Основна панель учня	Учень
adaptive-quiz	Модуль проходження тесту	Учень
teacher-dashboard	Керування питаннями	Вчитель
admin-dashboard	Керування користувачами	Адміністратор

Таблиця відображає компонентну структуру інтерфейсу вебзастосунку та розподіл відповідальності між модулями. Компонент dashboard-header забезпечує загальну навігацію та доступ до функції виходу і використовується всіма ролями. student-dashboard є основною панеллю учня, де відображається прогрес, XP та доступні навчальні активності. Модуль adaptive-quiz реалізує функціонал проходження адаптивного тесту. Для ролі вчителя передбачено teacher-dashboard, який дозволяє керувати банком питань, а admin-dashboard забезпечує адміністративні можливості, зокрема керування користувачами системи.

Отже, система гейміфікації та користувацький інтерфейс реалізують важливу мотиваційну та візуальну складову вебзастосунку. Поєднання механіки XP, рівнів та рейтингу з інтуїтивно зрозумілим інтерфейсом забезпечує підвищення залученості користувачів і сприяє ефективнішому навчанню.

3 РОЗРОБКА ТА ТЕСТУВАННЯ ВЕБЗАСТОСУНКУ

3.1 Реалізація програмного продукту та опис використаних технологій

Реалізація вебзастосунку виконана як цілісна серверно-клієнтська система на базі фреймворку Next.js 16, що дозволило поєднати інтерфейс користувача, бізнес-логіку та доступ до даних в межах єдиної архітектури. Основний акцент у реалізації зроблено на використанні Server Actions, які забезпечують виконання серверної логіки без створення окремого REST API, що значно спрощує структуру застосунку та підвищує його продуктивність.

У процесі розробки використано сучасний набір технологій, який забезпечує стабільність, масштабованість та зручність підтримки системи. Основні компоненти технологічного стеку наведено в таблиці 3.1.

Таблиця 3.1 – Технології, використані у вебзастосунку

Категорія	Технологія	Версія	Призначення
Фреймворк	Next.js [22]	16	Побудова вебзастосунку, маршрутизація, SSR, Server Actions
Мова програмування	TypeScript [23]	5.x	Типізація та підвищення надійності коду
Середовище виконання	Node.js [24]	24	Виконання серверної частини застосунку
База даних	SQLite (node:sqlite) [25]	вбудована	Локальне зберігання даних без окремого сервера
Безпека	Bcryptjs [26]	2.x	Хешування паролів користувачів
Стилізація	Tailwind CSS [27]	4.x	Побудова адаптивного інтерфейсу
UI-компоненти	shadcn/ui [28]	latest	Готові компоненти інтерфейсу
Іконки	lucide-react [29]	latest	Векторні іконки для UI

Таблиця відображає основні технології, використані при розробці вебзастосунку, їхні версії та призначення. У ній представлено ключові компоненти системи: фреймворк Next.js для побудови застосунку та серверної логіки, TypeScript для типізації коду, Node.js як середовище виконання, а також SQLite для зберігання даних. Окрім цього, зазначено інструменти безпеки

(bcryptjs), стилізації (Tailwind CSS), UI-бібліотеку (shadcn/ui) та набір іконок lucide-react, що забезпечують реалізацію інтерфейсу користувача.

Структура проєкту побудована за принципом розділення відповідальностей між каталогами, що забезпечує модульність, читабельність та простоту масштабування системи (рис. 3.1).

Діаграма відображає структуру проєкту вебзастосунку LinguaQuest та демонструє логічний розподіл файлів і директорій за функціональними модулями.

У центрі архітектури знаходиться каталог app/, який містить основні сторінки застосунку. Зокрема, директорія page.tsx реалізує лендинг-сторінку, тоді як login/ та register/ відповідають за процеси автентифікації користувачів – вхід та реєстрацію в системі. Окремо виділено каталог dashboard/, який містить інтерфейси користувачів різних ролей та забезпечує доступ до функціоналу системи після авторизації.

Файл layout.tsx виконує роль кореневого макета застосунку, забезпечуючи єдину структуру інтерфейсу для всіх сторінок.

Рівень роботи з даними представлений файлом db.ts, який відповідає за підключення до бази даних SQLite та визначення її структури. Файл auth.ts реалізує логіку автентифікації користувачів, тоді як levels.ts містить механізм нарахування XP та розрахунку рівнів.

Каталог actions/ містить серверні дії (Server Actions), які реалізують бізнес-логіку застосунку безпосередньо на сервері. Це дозволяє уникнути окремого REST API та спростити архітектуру системи.

Окремо виділено рольові модулі: student/ відповідає за функціонал учня, teacher/ – за інструменти викладача, а admin/ – за адміністративну частину системи, включаючи керування користувачами та даними.

Каталог ui/ містить базові багаторазові компоненти інтерфейсу, які використовуються у різних частинах застосунку для забезпечення єдиного стилю та повторного використання коду.

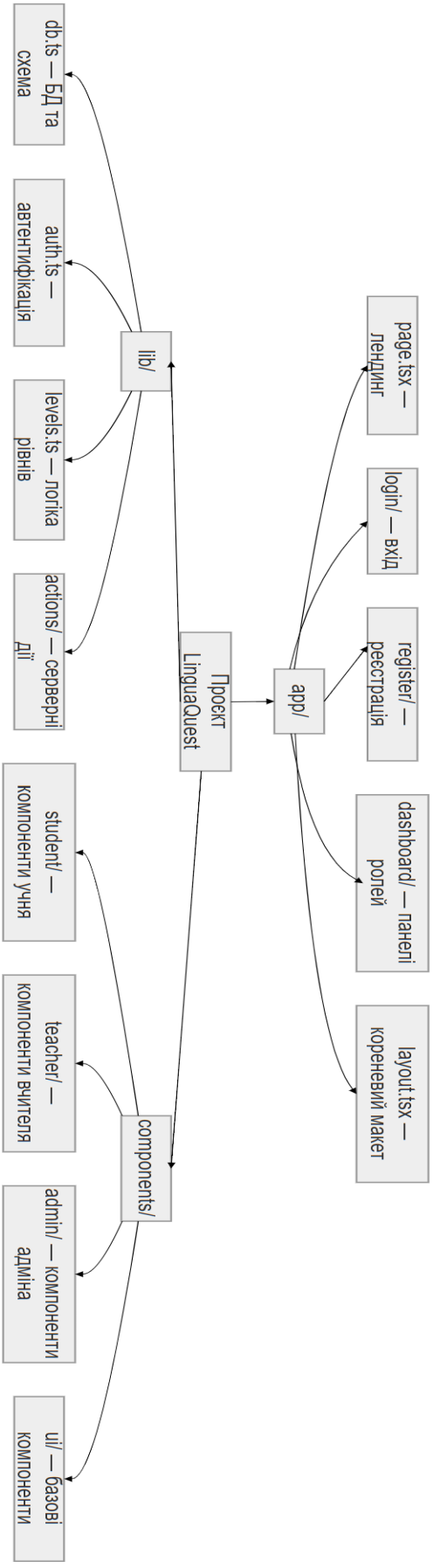


Рисунок 3.1 – Структура проєкту вебзастосунку LinguaQuest

Таким чином, подана структура демонструє модульний підхід до розробки, де кожен компонент системи відповідає за окрему функціональну зону, що забезпечує масштабованість та простоту підтримки проєкту.

Рівень доступу до даних у вебзастосунку реалізовано на основі вбудованого модуля node:sqlite, який входить до стандартного середовища Node.js 24 і дозволяє працювати з базою даних без необхідності використання окремого серверного рушія СУБД (додаток А.1). Такий підхід забезпечує простоту розгортання системи та зменшує кількість зовнішніх залежностей, що є особливо важливим для навчальних та MVP-рішень.

Підключення до бази даних організовано через функцію створення з’єднання, яка ініціалізує файл бази даних у кореневій директорії проєкту. Після створення з’єднання виконується початкова конфігурація SQLite, зокрема вмикається режим WAL (Write-Ahead Logging) для підвищення продуктивності при одночасному читанні та записі, а також активується підтримка зовнішніх ключів для забезпечення цілісності зв’язків між таблицями.

Додатково під час ініціалізації викликаються функції створення схеми бази даних та початкового наповнення тестовими або базовими даними. Це дозволяє автоматично підготувати середовище до роботи без ручного налаштування.

Окрему увагу приділено проблемі повторного створення підключення під час гарячого перезавантаження застосунку. Для вирішення цієї проблеми реалізовано кешування з’єднання у глобальному просторі змінних. Перед повторним використанням виконується перевірка валідності активного підключення шляхом виконання тестового SQL-запиту. Якщо з’єднання є неактуальним або пошкодженим, воно повторно ініціалізується.

Такий підхід забезпечує стабільність роботи застосунку в режимі розробки та запобігає дублюванню підключень до бази даних.

Оскільки об’єкти, які повертає node:sqlite, мають специфічну внутрішню структуру (рядки з null-прототипом), вони не можуть напряму використовуватися у клієнтських компонентах. Для вирішення цієї проблеми

реалізовано допоміжні функції-обгортки, які перетворюють результати запитів у звичайні JavaScript-об’єкти. Це дозволяє безпечно передавати дані між серверною та клієнтською частинами застосунку без втрати сумісності.

Таким чином, реалізований рівень доступу до даних забезпечує стабільну роботу з базою, контроль цілісності даних та коректну інтеграцію з серверною архітектурою Next.js.

Механізм автентифікації у вебзастосунку побудований на основі перевіреної схеми “email + пароль” із додатковим захистом у вигляді хешування паролів та використання сесійної авторизації через HTTP-only cookies (додаток А.2).

Під час процесу входу користувача система спочатку виконує пошук запису в базі даних за електронною поштою. Якщо користувача не знайдено або введений пароль не збігається із збереженим хешем, система повертає повідомлення про помилку автентифікації. Для захисту паролів використовується бібліотека bcryptjs, яка забезпечує їх необоротне хешування та перевірку у безпечний спосіб.

У разі успішної автентифікації створюється нова сесія користувача. Кожна сесія має унікальний ідентифікатор, який генерується випадковим чином (UUID). Цей ідентифікатор зберігається у таблиці сесій бази даних та одночасно передається клієнту у вигляді HTTP-only cookie.

Використання cookie з параметром HTTP-only забезпечує захист від доступу через JavaScript на стороні клієнта, що знижує ризик атак типу XSS. Додатково встановлюються параметри sameSite, path та термін дії cookie, який у даній реалізації становить 7 днів, що дозволяє підтримувати баланс між зручністю користувача та безпекою.

Таким чином, реалізована система автентифікації забезпечує перевірку користувачів, захищене зберігання облікових даних та стійкий механізм підтримки сесій без використання зовнішніх сервісів авторизації.

Отже, розглянуто процес реалізації вебзастосунку та обґрунтовано вибір основних технологій, що забезпечують його функціонування. Запропонований

технологічний стек на базі Next.js 16, TypeScript, Node.js 24 та SQLite дозволяє реалізувати цілісний монолітний застосунок із серверним рендерингом, вбудованою роботою з базою даних і мінімальною кількістю зовнішніх залежностей. Окрему увагу приділено архітектурній організації проєкту, що базується на поділі відповідальностей між модулями, а також реалізації ключових системних компонентів: доступу до даних, автентифікації та сесійного управління. Це забезпечує стабільність, масштабованість та підтримуваність програмного продукту.

Таким чином, реалізація програмного продукту демонструє ефективне поєднання сучасних вебтехнологій і практичних архітектурних рішень, що відповідають вимогам до навчальних адаптивних систем.

3.2 Реалізація функціоналу адаптивних тестів і персоналізованих рекомендацій

На серверній стороні реалізовано функцію отримання тестових питань із бази даних. Вона виконує вибірку всіх питань із таблиці questions та групує їх за рівнями складності для подальшої адаптивної обробки на клієнті.

```
export async function getQuizQuestions() {
  const rows = queryAll<QuestionRow>(
    "SELECT id, text, options, correct_index, difficulty, topic FROM questions",
  )
  const byDifficulty = { beginner: [], intermediate: [], advanced: [] }
  for (const r of rows) {
    byDifficulty[r.difficulty].push({
      id: r.id,
      text: r.text,
      options: JSON.parse(r.options),
```

```
      correctIndex: r.correct_index,
      difficulty: r.difficulty,
      topic: r.topic,
    })
  }
  return byDifficulty
}
```

Дана серверна функція відповідає за формування набору тестових питань. Спочатку виконується SQL-запит до таблиці questions, після чого отримані записи групуються за рівнями складності: beginner, intermediate та advanced. Поле options зберігається у форматі JSON і перетворюється у звичайний масив. Важливо, що правильна відповідь (correctIndex) не використовується для логіки на клієнті до моменту перевірки, що підвищує захищеність системи від підглядання через інструменти розробника.

Адаптивність тесту реалізовано через механізм зміни складності наступного питання залежно від правильності відповіді користувача. Для цього визначено набір рівнів складності та функцію переходу між ними.

```
const LEVELS = ["beginner", "intermediate", "advanced"] as const
type Difficulty = (typeof LEVELS)[number]
function nextDifficulty(current: Difficulty, isCorrect: boolean): Difficulty {
  const idx = LEVELS.indexOf(current)
  if (isCorrect) {
    return LEVELS[Math.min(idx + 1, LEVELS.length - 1)]
  }
  return LEVELS[Math.max(idx - 1, 0)]
}
```

Функція nextDifficulty реалізує логіку адаптивного тестування. Якщо користувач відповідає правильно, рівень складності підвищується, якщо неправильно – знижується. Межі масиву враховуються, щоб рівень не виходив за допустимі значення. Таким чином система динамічно підлаштовує складність тесту під рівень знань користувача.

Для стимулювання навчальної активності реалізовано систему нарахування балів досвіду (XP), яка залежить від складності питання.

```
const XP_BY_DIFFICULTY: Record<Difficulty, number> = {
  beginner: 10,
  intermediate: 20,
  advanced: 30,
}
```

Кожному рівню складності відповідає певна кількість балів XP. Чим складніше питання, тим більше балів отримує користувач за правильну відповідь, що мотивує проходження більш складних завдань.

Обробка відповіді користувача виконується безпосередньо в компоненті тесту:

```
function handleAnswer(selectedIndex: number) {
  const correct = selectedIndex === question.correctIndex
  if (correct) {
    setEarnedXp((xp) => xp + XP_BY_DIFFICULTY[difficulty])
    setCorrectCount((c) => c + 1)
  }
  setDifficulty((d) => nextDifficulty(d, correct))
  setAnswered(true)
}
```

Функція handleAnswer визначає, чи правильна відповідь була обрана користувачем, порівнюючи індекс вибраного варіанту з правильним. У разі правильної відповіді нараховується XP та збільшується лічильник правильних відповідей. Після цього викликається функція зміни складності, яка формує наступне питання. Також встановлюється стан answered, який блокує повторну відповідь і дозволяє показати користувачу результат.

Після завершення тесту результати зберігаються в базі даних, а також оновлюється загальний прогрес користувача.

```
export async function submitTestResult(data: {
  total: number
  correct: number
  xpEarned: number
  finalLevel: number
}) {
  const user = await getCurrentUser()
  if (!user) return { error: "Не авторизовано" }
  db.prepare(
    `INSERT INTO test_results (user_id, total, correct, xp_earned, final_level)
     VALUES (?, ?, ?, ?, ?)`
  ).run(user.id, data.total, data.correct, data.xpEarned, data.finalLevel)
  db.prepare(
    "UPDATE users SET xp = xp + ? WHERE id = ?"
  ).run(data.xpEarned, user.id)
  revalidatePath("/dashboard")
  return { ok: true }
}
```

Ця серверна функція виконує дві основні операції: збереження результатів тестування в таблицю test_results та оновлення загального XP користувача.

Використання двох SQL-запитів дозволяє розділити історію проходження тестів і поточний прогрес користувача. Після оновлення даних виконується ревалідація сторінки дашборду, щоб користувач одразу бачив актуальні результати.

На основі результатів тестування система формує індивідуальні рекомендації для подальшого навчання користувача.

```
function getRecommendation(correct: number, total: number): string {
  const ratio = total > 0 ? correct / total : 0
  if (ratio <= 0.4) {
    return "Повторіть базову граматику початкового рівня"
  }
  if (ratio <= 0.7) {
    return "Закріплюйте середній рівень, переходьте до складніших тем"
  }
  return "Відмінно! Спробуйте завдання просунутого рівня"
}
```

Функція аналізує результат тесту у вигляді частки правильних відповідей. На основі цього значення система визначає рівень підготовки користувача та формує рекомендацію. Якщо результат низький, пропонується повторення базових тем, при середньому результаті – закріплення матеріалу, а при високому – перехід до складніших завдань. Такий підхід забезпечує персоналізацію навчального процесу та адаптацію контенту під кожного користувача.

Отже, реалізовано ключовий функціонал вебзастосунку, що забезпечує адаптивне тестування та формування персоналізованих навчальних рекомендацій. Розроблений механізм поєднує серверну вибірку питань, клієнтську логіку динамічної зміни складності та систему нарахування балів досвіду (XP), що дозволяє автоматично підлаштовувати рівень завдань під користувача.

Окремо реалізовано збереження результатів тестування та оновлення прогресу користувача в базі даних, а також алгоритм формування рекомендацій на основі відсотка правильних відповідей. Це забезпечує не лише оцінювання знань, а й індивідуалізацію подальшого навчального процесу.

Таким чином, розроблений функціонал є основою персоналізованої освітньої системи, яка підвищує ефективність навчання за рахунок адаптації складності завдань і рекомендацій під рівень підготовки кожного користувача.

3.3 Реалізація ігрових елементів та системи мотивації користувачів

Систему мотивації побудовано на балах досвіду (XP) та автоматично розрахованих рівнях. Логіку винесено в окремий модуль levels.ts:

```
const XP_PER_LEVEL = 100
export function getLevel(xp: number): number {
  return Math.floor(xp / XP_PER_LEVEL) + 1
}
export function getLevelProgress(xp: number): number {
  return (xp % XP_PER_LEVEL) / XP_PER_LEVEL * 100  // % до наступного рівня
}
export function getXpForNextLevel(xp: number): number {
  return XP_PER_LEVEL - (xp % XP_PER_LEVEL)            // скільки XP залишилось
}
const LEVEL_TITLES = ["Початківець", "Учень", "Впевнений", "Досвідчений", "Майстер"]
export function getLevelTitle(level: number): string {
  return LEVEL_TITLES[Math.min(level - 1, LEVEL_TITLES.length - 1)]
```

```
}
```

Логіка цього модуля базується на простій прогресійній моделі, де кожні 100 XP формують один рівень. Функція getLevel визначає поточний рівень користувача шляхом цілочисельного ділення XP на фіксований поріг. Функція getLevelProgress повертає відсоток заповнення поточного рівня, що використовується для візуального прогрес-бару. getXpForNextLevel показує залишок XP до наступного рівня, що підсилює мотиваційний ефект. Додатково getLevelTitle призначає кожному рівню умовну назву, що підвищує ігрову складову системи.

У панелі учня прогрес користувача формується на основі його поточного XP та візуалізується у вигляді інформаційної картки.

```
const level = getLevel(user.xp)
const progress = getLevelProgress(user.xp)
const remaining = getXpForNextLevel(user.xp)
return (
  <Card>
    <CardHeader>
      <CardTitle>Рівень {level} – {getLevelTitle(level)}</CardTitle>
    </CardHeader>
    <CardContent>
      <div className="flex justify-between text-sm">
        <span>{user.xp} XP</span>
        <span>До наступного рівня: {remaining} XP</span>
      </div>
      <Progress value={progress} className="mt-2" />
    </CardContent>
  </Card>
)
```

Цей компонент відповідає за відображення ключових показників прогресу користувача. На основі функцій із модуля levels.ts обчислюється рівень, відсоток заповнення поточного рівня та залишок XP до наступного. UI-компонент поєднує текстову інформацію з прогрес-баром, що дозволяє користувачу візуально оцінювати власний розвиток у системі.

Для підсилення мотиваційного ефекту реалізовано рейтинг користувачів, який базується на загальній кількості XP.

```
export async function getLeaderboard(limit = 10): Promise<LeaderboardEntry[]> {
  return queryAll<LeaderboardEntry>(
    "SELECT id, name, xp FROM users WHERE role = ‘student’ ORDER BY xp DESC LIMIT ?",
    limit,
  )
}
```

Даний запит отримує список користувачів із роллю студента, сортує їх за спаданням XP та обмежує кількість записів відповідно до параметра limit. Таким чином формується таблиця лідерів, яка оновлюється динамічно та відображає найактивніших користувачів системи.

Логіка взаємодії ігрових елементів наведена на рис. 3.2.

Діаграма відображає повний цикл роботи системи гейміфікації. Після проходження тесту користувач отримує XP, які записуються у базу даних. На основі оновленого значення XP автоматично перераховується рівень та прогрес до наступного рівня. Паралельно оновлюється таблиця лідерів, що дозволяє користувачу бачити своє місце серед інших учасників. Сукупність цих процесів створює замкнений мотиваційний цикл, який стимулює подальше навчання.

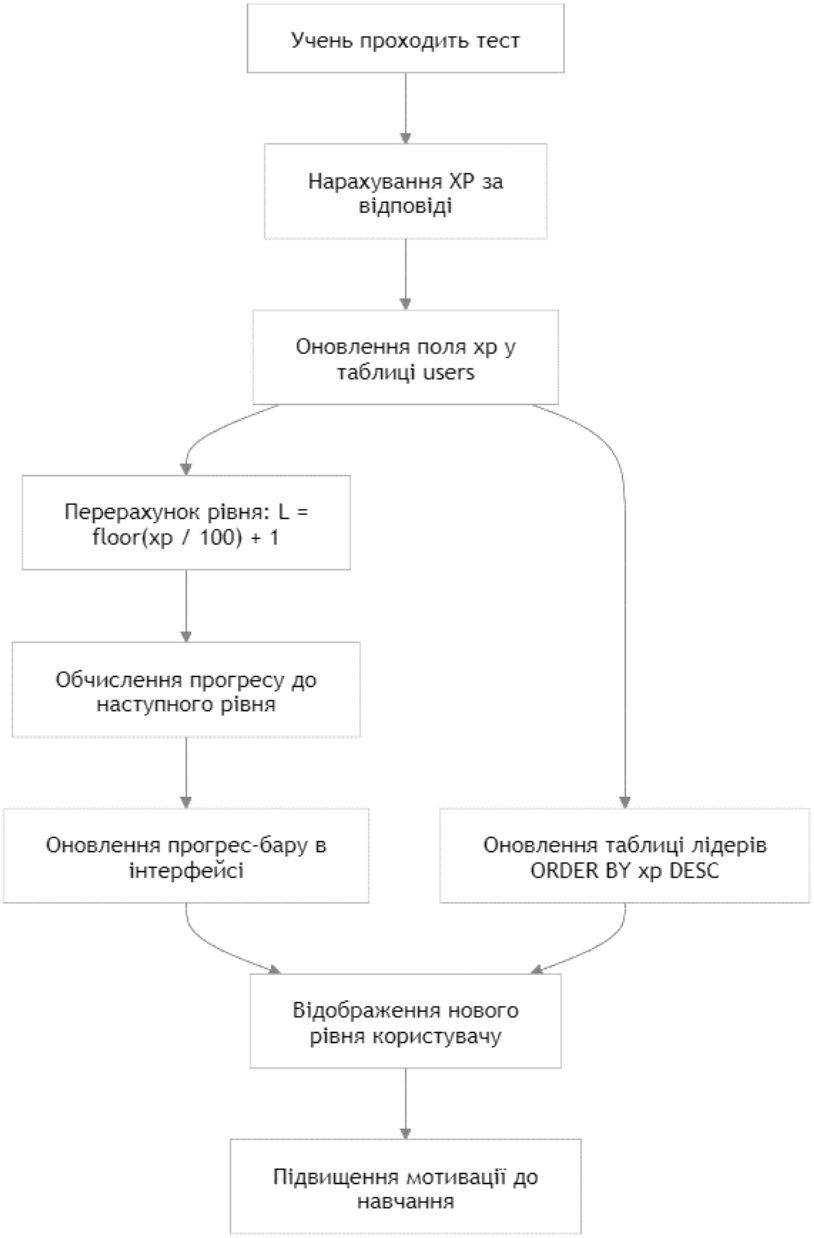


Рисунок 3.2 – Логіка взаємодії ігрових елементів

Отже, реалізовано систему ігрових елементів, що базується на механіці нарахування XP, автоматичному розрахунку рівнів та візуалізації прогресу користувача. Додатково впроваджено таблицю лідерів, яка формує елемент змагання між учасниками та підвищує рівень залученості.

Запропонована система забезпечує безперервний мотиваційний цикл: виконання завдань – отримання XP – підвищення рівня – візуальний прогрес – порівняння результатів із іншими користувачами. Це сприяє підвищенню активності користувачів та регулярності навчального процесу.

3.4 Тестування вебзастосунку та аналіз результатів його роботи

Тестування вебзастосунку було спрямоване на перевірку коректності роботи всіх функціональних модулів системи, а також відповідності реалізованої логіки вимогам технічного завдання. Основним підходом обрано функціональне ручне тестування, яке дозволяє оцінити поведінку системи з точки зору кінцевого користувача, а також додатково перевірити серверну логіку обробки даних і взаємодії з базою даних. Окрему увагу приділено перевірці сценаріїв для різних ролей користувачів – учня, вчителя та адміністратора, оскільки саме рольова модель визначає доступ до основного функціоналу системи.

Тестування виконувалося у середовищі Node.js 24 із використанням браузера на базі Chromium, що забезпечило повну сумісність із реалізованими технологіями, зокрема Server Actions та клієнтськими компонентами React. У процесі перевірки особливу увагу приділяли коректності маршрутизації, обробці авторизації, роботі адаптивного тестування, нарахуванню XP, а також оновленню рівнів користувачів після завершення тестів.

Для проведення тестування були створені три тестові облікові записи, які відповідають основним ролям системи: адміністратор, вчитель та учень. Такий підхід дозволив перевірити розмежування доступу та переконатися, що кожна роль має доступ лише до передбаченого їй функціоналу. Зокрема, адміністратор перевірявся на можливість керування користувачами та перегляд загальної статистики, вчитель – на створення та редагування навчальних матеріалів, а учень – на проходження адаптивних тестів і отримання результатів.

Функціональне тестування охопило повний життєвий цикл роботи користувача в системі – від входу та реєстрації до завершення тестування і виходу з акаунта. Усі основні тест-кейси, що перевіряли авторизацію, створення користувачів, запуск тестів, нарахування XP, зміну складності питань, збереження результатів та роботу адміністративної панелі, були успішно

виконані. Це підтвердило стабільність роботи системи та коректну реалізацію бізнес-логіки.

Окремо проводилося тестування адаптивного механізму зміни складності запитань. У ході перевірки моделювалися різні сценарії відповідей користувача, включаючи правильні та неправильні відповіді на різних рівнях складності. Результати показали, що система коректно підвищує складність при успішних відповідях і знижує її у разі помилок, при цьому не виходячи за межі допустимих рівнів. Це підтверджує правильність реалізації адаптивного алгоритму навчання.

На основі отриманих результатів також було проведено аналіз виявлених проблем. Зокрема, на етапі розробки виникала проблема передачі даних у клієнтську частину через особливості роботи node:sqlite, який повертає об’єкти з null-прототипом. Це було вирішено шляхом створення допоміжних функцій для нормалізації даних перед їх передачею на клієнт. Також було усунуто проблему нестабільного підключення до бази даних шляхом впровадження кешування з перевіркою валідності з’єднання. Загалом, після внесених виправлень система почала працювати стабільно без критичних помилок.

Підсумковий аналіз результатів тестування показав, що всі основні модулі вебзастосунку функціонують коректно, а логіка взаємодії між компонентами є узгодженою та стабільною. Це дозволяє зробити висновок про готовність системи до використання в умовах реальної експлуатації.

3.5 Інструкція користувача

Вебзастосунок побудований таким чином, щоб забезпечити просту та інтуїтивно зрозумілу взаємодію для всіх категорій користувачів, тому процес початку роботи є максимально спрощеним. Користувач відкриває головну сторінку системи, де йому доступні основні інформаційні блоки та варіанти подальших дій – вхід у систему або створення нового облікового запису. Після вибору відповідної дії користувач або проходить авторизацію, або реєструється,

після чого автоматично потрапляє до особистого кабінету відповідно до своєї ролі.

Подальша робота користувача залежить від його ролі в системі. Учень після входу отримує доступ до персональної панелі, де відображається його поточний рівень, кількість накопичених XP, прогрес до наступного рівня, а також позиція в рейтингу. Основним елементом взаємодії є проходження адаптивних тестів, які змінюють рівень складності питань залежно від правильності відповідей. Після завершення тестування користувач отримує детальний результат, що включає кількість правильних відповідей, отримані XP та персоналізовану рекомендацію щодо подальшого навчання. Також користувач може відстежувати свій прогрес через таблицю лідерів та історію проходження тестів, що підсилює мотиваційний ефект системи.

Вчитель взаємодіє з системою через окрему панель керування навчальним контентом. Його основна задача полягає у створенні тестових питань, які включають текст запитання, варіанти відповідей, правильний варіант, рівень складності та тематичну категорію. Після збереження такі питання автоматично потрапляють до загального банку та використовуються у формуванні адаптивних тестів, що забезпечує постійне оновлення навчального контенту.

Адміністратор має розширені права доступу до системи та відповідає за управління користувачами та загальною статистикою платформи. Через адміністративну панель він може переглядати кількість користувачів, контролювати активність системи, змінювати ролі облікових записів або видаляти їх у разі потреби. Це забезпечує повний контроль над функціонуванням вебзастосунку та його користувацькою базою.

Завершення роботи в системі реалізовано через функцію виходу, яка доступна у верхній навігаційній панелі. Після її активації сесія користувача автоматично завершується, дані авторизації видаляються, а користувач перенаправляється на головну сторінку застосунку. Такий підхід забезпечує безпеку даних і коректне завершення роботи із системою.

ВИСНОВКИ

У межах кваліфікаційної роботи було здійснено комплексне дослідження, проєктування, розробку та тестування вебзастосунку для персоналізованого навчання іноземних мов, що поєднує адаптивні навчальні механізми та елементи гейміфікації. Отримані результати дозволяють зробити узагальнені висновки відповідно до структури виконаних завдань.

На етапі теоретичного дослідження було розглянуто особливості застосування інформаційних технологій у сфері вивчення іноземних мов. Встановлено, що сучасні цифрові освітні платформи активно переходять від стандартного подання матеріалу до персоналізованих моделей навчання, які враховують індивідуальний рівень знань, темп засвоєння матеріалу та поведінкові особливості користувача. Проведений аналіз існуючих платформ показав, що найбільш ефективними є рішення, які поєднують адаптивне тестування, інтерактивні вправи та систему мотивації через ігрові елементи. На основі цього були сформульовані функціональні та нефункціональні вимоги до розроблюваного вебзастосунку, зокрема щодо ролей користувачів, безпеки даних, продуктивності та зручності інтерфейсу.

У процесі проєктування системи було виконано аналіз предметної області та розроблено архітектуру вебзастосунку, що базується на сучасному підході до побудови серверно-клієнтних рішень. Окрему увагу приділено проєктуванню бази даних, яка забезпечує збереження інформації про користувачів, тестові завдання, результати проходження тестів та рівні прогресу. Також було спроєктовано модульну структуру застосунку, що дозволяє розділити функціональність за ролями користувачів. Важливим етапом стало проєктування механізму адаптивного тестування, який забезпечує динамічну зміну складності завдань залежно від результатів користувача, а також формування індивідуальної траєкторії навчання. Додатково було розроблено концепцію

системи гейміфікації та інтерфейсу користувача, спрямованих на підвищення мотивації до навчання.

На етапі реалізації було створено повнофункціональний вебзастосунок із використанням сучасного технологічного стеку, що забезпечує високу продуктивність та масштабованість системи. Реалізовано адаптивні тести, які змінюють рівень складності в реальному часі, а також систему персоналізованих рекомендацій, що формуються на основі результатів користувача. Окремо впроваджено гейміфікаційні елементи, зокрема систему рівнів, нарахування XP та рейтинг користувачів, що сприяє підвищенню залученості. Проведене тестування підтвердило коректність роботи всіх основних модулів системи, стабільність її функціонування та відповідність визначеним вимогам. У ході тестування також було виявлено та усунуто окремі технічні проблеми, що дозволило підвищити надійність застосунку.

Таким чином, розроблений вебзастосунок реалізує концепцію персоналізованого навчання іноземних мов із використанням адаптивних алгоритмів та елементів гейміфікації. Отримані результати підтверджують ефективність запропонованих підходів і можуть бути використані для подальшого розвитку подібних освітніх систем, зокрема шляхом розширення функціоналу, впровадження більш складних моделей адаптації та інтеграції штучного інтелекту для аналізу навчальної поведінки користувачів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Василик М. С., Яциняк О. П., Орендарчук О. Л. Використання цифрових технологій у навчанні іноземним мовам: дослідження ефективності та викликів. *Педагогічна Академія: наукові записки.* 2025. № 17. URL : https://doi.org/10.5281/zenodo.15213717 (дата звернення : 05.04.2026).

2. Колонуто Д. К. Сучасні технології у викладанні іноземних мов. *Науковий вісник Ужгородського національного університету. Серія "Педагогіка, соціальна робота".* 2014. Вип. 33. С. 92–94.

3. Шийка С. В. Інформаційно-комунікаційні технології як засіб удосконалення традиційної методики навчання української мови за професійним спрямуванням. *Інформаційні технології і засоби навчання*. 2017. Т. 59, № 3. С. 114–125.

4. Грабовська Н. А., Цимбалюк Т. П. Інформаційні технології у вивченні іноземної мови. *Інноваційна педагогіка*. 2020. Вип. 24. Т. 2. С. 222–225.

5. Cotocus Blog. Top 10 Language Learning Apps in 2025: Features, Pros, Cons & Comparison. URL : https://www.cotocus.com/blog/top-10-language-learning-apps-in-2025-features-pros-cons-comparison/ (дата звернення : 05.04.2026).

6. Harvard Business Review. How Duolingo Aims to Diversify Beyond Language Learning. URL : https://hbr.org/podcast/2025/04/how-duolingo-aims-to-diversify-beyond-language-learning (дата звернення : 05.04.2026).

7. IEEE Spectrum. How Duolingo's AI Learns What You Need to Learn. URL : https://spectrum.ieee.org/duolingo (дата звернення : 05.04.2026).

8. Duolingo. Duolingo Introduces AI-Powered Innovations at Duocon 2024. URL : https://investors.duolingo.com/news-releases/news-release-details/duolingo-introduces-ai-powered-innovations-duocon-2024 (дата звернення : 05.04.2026).

9. ScienceDirect. Effects of two mobile-assisted language learning apps on L2 receptive and productive vocabulary knowledge: A mixed-methods study. URL :

https://www.sciencedirect.com/science/article/abs/pii/S0346251X25001733 (дата звернення : 05.04.2026).

10. Krioda. Busuu vs Babbel: A Detailed Comparison. URL : https://www.krioda.com/busuu-vs-babbel/ (дата звернення : 05.04.2026).

11. Taalhammer. Which Language Learning App Is Best for Personalised Learning in 2026? URL : https://www.taalhammer.com/which-language-learning-app-is-best-for-personalised-learning-in-2026-taalhammer-vs-duolingo-busuu-babbel-anki-and-memrise/ (дата звернення : 05.04.2026).

12. Talkpal AI. Busuu vs Babbel: Which Language App Wins in 2024? URL : https://talkpal.ai/busuu-vs-babbel-which-language-app-wins-in-2024/ (дата звернення : 05.04.2026).

13. Hello Nabu. 5 Best Language Learning Apps in 2026. URL : https://www.hellonabu.com/blog/en/best-language-learning-apps/ (дата звернення : 05.04.2026).

14. Frontiers in Education. Recent developments in mobile-assisted vocabulary learning: a mini review of published studies focusing on digital flashcards. Frontiers in Education. 2024. Vol. 9. DOI : https://doi.org/10.3389/feduc.2024.1496578 (дата звернення : 05.04.2026).

15. Talkreal. Mondly vs. Rosetta Stone: The Battle of Language Learning. URL : https://talkreal.org/en/blog/mondly-vs-rosetta-stone/ (дата звернення : 05.04.2026).

16. Krioda. 15 Language Learning Apps Compared – Personally Tested. URL : https://www.krioda.com/language-learning-apps-compared/ (дата звернення : 05.04.2026).

17. Lingoda. Online vs offline language schools: How to choose? URL : https://www.lingoda.com/blog/en/online-vs-offline-language-schools/ (дата звернення : 05.04.2026).

18. Lingoda. Lingoda vs. Rosetta Stone: Which language learning platform fits your goals? URL : https://www.lingoda.com/blog/en/lingoda-vs-rosetta-stone/ (дата звернення : 05.04.2026).

19. Вінницький державний педагогічний університет. Адаптивне навчання у фаховій передвищій освіті в умовах цифровізації. Сучасні інформаційні технології та інноваційні методики навчання. 2024. Вип. 71. С. 245–252. URL : https://vspu.net/sit/index.php/sit/article/download/5749/5140 (дата звернення : 05.04.2026).

20. StudyGuides. Top 10 New Study Apps to Watch in 2025. URL : https://studyguides.com/articles/top-new-study-apps-to-watch (дата звернення : 05.04.2026).

21. Interconf. Переваги та недоліки адаптивного навчання іноземних мов. URL : https://archive.interconf.center/index.php/conference-proceeding/article/download/1066/1092 (дата звернення : 05.04.2026).

22. Next.js by Vercel – the React framework. URL : https://nextjs.org/ (дата звернення : 05.04.2026).

23. TypeScript: JavaScript with Syntax for Types. URL : https://www.typescriptlang.org/ (дата звернення : 05.04.2026).

24. Node.js – Run JavaScript everywhere. URL : https://nodejs.org/en (дата звернення : 05.04.2026).

25. SQLite Documentation. URL : https://sqlite.org/docs.html (дата звернення : 05.04.2026).

26. npm: bcryptjs. URL : https://www.npmjs.com/package/bcryptjs (дата звернення : 05.04.2026).

27. Tailwind CSS Documentation. URL : https://tailwindcss.com/ (дата звернення : 05.04.2026).

28. shadcn/ui Documentation. URL : https://ui.shadcn.com/ (дата звернення : 05.04.2026).

29. Lucide Icons. URL : https://lucide.dev/guide/react/ (дата звернення : 05.04.2026).

ДОДАТОК А

ЛІСТИНГ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ПРОДУКТУ

Лістинг А.1 – Реалізація рівня доступу до даних

```
import { DatabaseSync } from "node:sqlite"
import bcrypt from "bcryptjs"
import path from "path"

// Singleton-підключення до SQLite, щоб уникнути повторної ініціалізації під час HMR
declare global {
  // eslint-disable-next-line no-var
  var __linguaDb: DatabaseSync | undefined
}

function createConnection(): DatabaseSync {
  const dbPath = path.join(process.cwd(), "lingua.db")
  const db = new DatabaseSync(dbPath)
  db.exec("PRAGMA journal_mode = WAL;")
  db.exec("PRAGMA foreign_keys = ON;")
  initSchema(db)
  seed(db)
  return db
}

function getDb(): DatabaseSync {
```

```
  if (global.__linguaDb) {
    try {
      // Перевіряємо, що з'єднання валідне та схема існує
      global.__linguaDb.prepare("SELECT 1 FROM users LIMIT 1").get()
      return global.__linguaDb
    } catch {
      // Кешоване з'єднання застаріло – створюємо нове
      try {
        global.__linguaDb.close()
      } catch {
        /* ignore */
      }
      global.__linguaDb = undefined
    }
  }
  const conn = createConnection()
  global.__linguaDb = conn
  return conn
}

export const db: DatabaseSync = getDb()

// node:sqlite повертає рядки з null-прототипом, які не можна передавати
// у Client Components. Ці хелпери конвертують результати у звичайні об'єкти.
export function queryAll<T = Record<string, unknown>>(
  sql: string,
  ...params: unknown[]
): T[] {
```

```
  const rows = db.prepare(sql).all(...(params as never[])) as Record<string, unknown>[]
  return rows.map((r) => ({ ...r })) as T[]
}

export function queryOne<T = Record<string, unknown>>(
  sql: string,
  ...params: unknown[]
): T | undefined {
  const row = db.prepare(sql).get(...(params as never[])) as Record<string, unknown> | undefined
  return row ? ({ ...row } as T) : undefined
}

// Ролі застосунку
export type Role = "student" | "teacher" | "admin"

// Рівні складності питань (CEFR-подібні)
export const DIFFICULTY_LEVELS = ["beginner", "intermediate", "advanced"] as const
export type Difficulty = (typeof DIFFICULTY_LEVELS)[number]

function initSchema(db: DatabaseSync) {
  db.exec(`
    CREATE TABLE IF NOT EXISTS users (
      id INTEGER PRIMARY KEY AUTOINCREMENT,
      name TEXT NOT NULL,
      email TEXT NOT NULL UNIQUE,
      password TEXT NOT NULL,
      role TEXT NOT NULL CHECK (role IN ('student','teacher','admin')),
```

```
    xp INTEGER NOT NULL DEFAULT 0,
    created_at TEXT NOT NULL DEFAULT (datetime('now'))
  );

  CREATE TABLE IF NOT EXISTS questions (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    prompt TEXT NOT NULL,
    option_a TEXT NOT NULL,
    option_b TEXT NOT NULL,
    option_c TEXT NOT NULL,
    option_d TEXT NOT NULL,
    correct_index INTEGER NOT NULL,
    difficulty TEXT NOT NULL CHECK (difficulty IN ('beginner','intermediate','advanced')),
    topic TEXT NOT NULL DEFAULT 'General',
    author_id INTEGER,
    created_at TEXT NOT NULL DEFAULT (datetime('now')),
    FOREIGN KEY (author_id) REFERENCES users(id) ON DELETE SET NULL
  );

  CREATE TABLE IF NOT EXISTS sessions (
    id TEXT PRIMARY KEY,
    user_id INTEGER NOT NULL,
    created_at TEXT NOT NULL DEFAULT (datetime('now')),
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
  );

  CREATE TABLE IF NOT EXISTS test_results (
```

```
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        user_id INTEGER NOT NULL,
        total INTEGER NOT NULL,
        correct INTEGER NOT NULL,
        xp_earned INTEGER NOT NULL,
        final_level TEXT NOT NULL,
        created_at TEXT NOT NULL DEFAULT (datetime('now')),
        FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
      );
    `)
  }

  function seed(db: DatabaseSync) {
    const count = (db.prepare("SELECT COUNT(*) AS c FROM users").get() as { c: number }).c
    if (count > 0) return

    const hash = (pw: string) => bcrypt.hashSync(pw, 10)
    const insertUser = db.prepare(
      "INSERT INTO users (name, email, password, role, xp) VALUES (?, ?, ?, ?, ?)",
    )
    insertUser.run("Адміністратор", "admin@lingua.app", hash("admin123"), "admin", 0)
    const teacherInfo = insertUser.run("Олена Вчителька", "teacher@lingua.app", hash("teacher123"), "teacher", 0)
    insertUser.run("Іван Учень", "student@lingua.app", hash("student123"), "student", 120)
```

```
const teacherId = Number(teacherInfo.lastInsertRowid)

const insertQ = db.prepare(
  `INSERT INTO questions (prompt, option_a, option_b, option_c, option_d, correct_index, difficulty, topic, author_id)
   VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)`,
)

const seedQuestions: Array<
  [string, string, string, string, string, number, Difficulty, string]
> = [
  // beginner
  ["Choose the correct article: ___ apple a day keeps the doctor away.", "A", "An", "The", "No article", 1, "beginner", "Grammar"],
  ["What is the plural of ‘child’?", "childs", "childes", "children", "childrens", 2, "beginner", "Vocabulary"],
  ["She ___ to school every day.", "go", "goes", "going", "gone", 1, "beginner", "Grammar"],
  ["Which word means a place where you sleep?", "kitchen", "bedroom", "garden", "garage", 1, "beginner", "Vocabulary"],
  ["I ___ a student.", "am", "is", "are", "be", 0, "beginner", "Grammar"],
  // intermediate
  ["If I ___ rich, I would travel the world.", "am", "was", "were", "be", 2, "intermediate", "Grammar"],
  ["Choose the synonym of ‘rapid’.", "slow", "quick", "lazy", "calm", 1, "intermediate", "Vocabulary"],
  ["She has lived here ___ 2010.", "for", "since", "from", "during", 1, "intermediate", "Grammar"],
  ["The opposite of ‘generous’ is:", "kind", "stingy", "happy", "brave", 1, "intermediate", "Vocabulary"],
```

```
["By the time we arrived, the film ___.", "started", "has started", "had started", "starts", 2, "intermediate", "Grammar"],
// advanced
["Hardly ___ the meeting begun when the fire alarm rang.", "had", "has", "did", "was", 0, "advanced", "Grammar"],
["Choose the word closest in meaning to ‘ubiquitous’.", "rare", "omnipresent", "hidden", "ancient", 1, "advanced", "Vocabulary"],
["The committee was criticized for its ___ approach to reform.", "lackadaisical", "energetic", "transparent", "decisive", 0, "advanced", "Vocabulary"],
["Not only ___ late, but he also forgot the documents.", "he was", "was he", "he is", "is he", 1, "advanced", "Grammar"],
["Were it not for your help, I ___ failed.", "will have", "would have", "had", "have", 1, "advanced", "Grammar"],
]

for (const q of seedQuestions) {
  insertQ.run(q[0], q[1], q[2], q[3], q[4], q[5], q[6], q[7], teacherId)
}
}
```

Лістинг А.2 – Реалізація автентифікації та сесій

```
import "server-only"
import { cookies } from "next/headers"
import { randomUUID } from "crypto"
import bcrypt from "bcryptjs"
import { db, queryOne, type Role } from "./db"

const COOKIE_NAME = "lingua_session"
```

```
export interface User {
  id: number
  name: string
  email: string
  role: Role
  xp: number
  created_at: string
}

interface UserRow extends User {
  password: string
}

export async function register(
  name: string,
  email: string,
  password: string,
  role: Role,
): Promise<{ ok: boolean; error?: string }> {
  const existing = db.prepare("SELECT id FROM users WHERE email = ?").get(email)
  if (existing) return { ok: false, error: "Користувач з таким email вже існує" }

  const hashed = bcrypt.hashSync(password, 10)
  const info = db
    .prepare("INSERT INTO users (name, email, password, role) VALUES (?, ?, ?, ?)")
    .run(name, email, hashed, role)
```

```
  await createSession(Number(info.lastInsertRowid))
  return { ok: true }
 }

 export async function login(
  email: string,
  password: string,
 ): Promise<{ ok: boolean; error?: string }> {
  const row = queryOne<UserRow>("SELECT * FROM users WHERE email =
?", email)
  if (!row) return { ok: false, error: "Невірний email або пароль" }
  if (!bcrypt.compareSync(password, row.password)) {
   return { ok: false, error: "Невірний email або пароль" }
  }
  await createSession(row.id)
  return { ok: true }
 }

 async function createSession(userId: number) {
  const sessionId = randomUUID()
  db.prepare("INSERT INTO sessions (id, user_id) VALUES (?,
?)").run(sessionId, userId)
  const cookieStore = await cookies()
  cookieStore.set(COOKIE_NAME, sessionId, {
   httpOnly: true,
   sameSite: "lax",
   secure: process.env.NODE_ENV === "production",
   path: "/",
   maxAge: 60 * 60 * 24 * 7,
  })
```

```
}

export async function logout() {
  const cookieStore = await cookies()
  const sessionId = cookieStore.get(COOKIE_NAME)?.value
  if (sessionId) {
    db.prepare("DELETE FROM sessions WHERE id = ?").run(sessionId)
    cookieStore.delete(COOKIE_NAME)
  }
}

export async function getCurrentUser(): Promise<User | null> {
  const cookieStore = await cookies()
  const sessionId = cookieStore.get(COOKIE_NAME)?.value
  if (!sessionId) return null

  const row = queryOne<User>(
    `SELECT u.id, u.name, u.email, u.role, u.xp, u.created_at
       FROM sessions s JOIN users u ON u.id = s.user_id
       WHERE s.id = ?`,
    sessionId,
  )

  return row ?? null
}
```

ДОДАТОК Б

ІНТЕРФЕЙС ВЕБЗАСТОСУНКУ

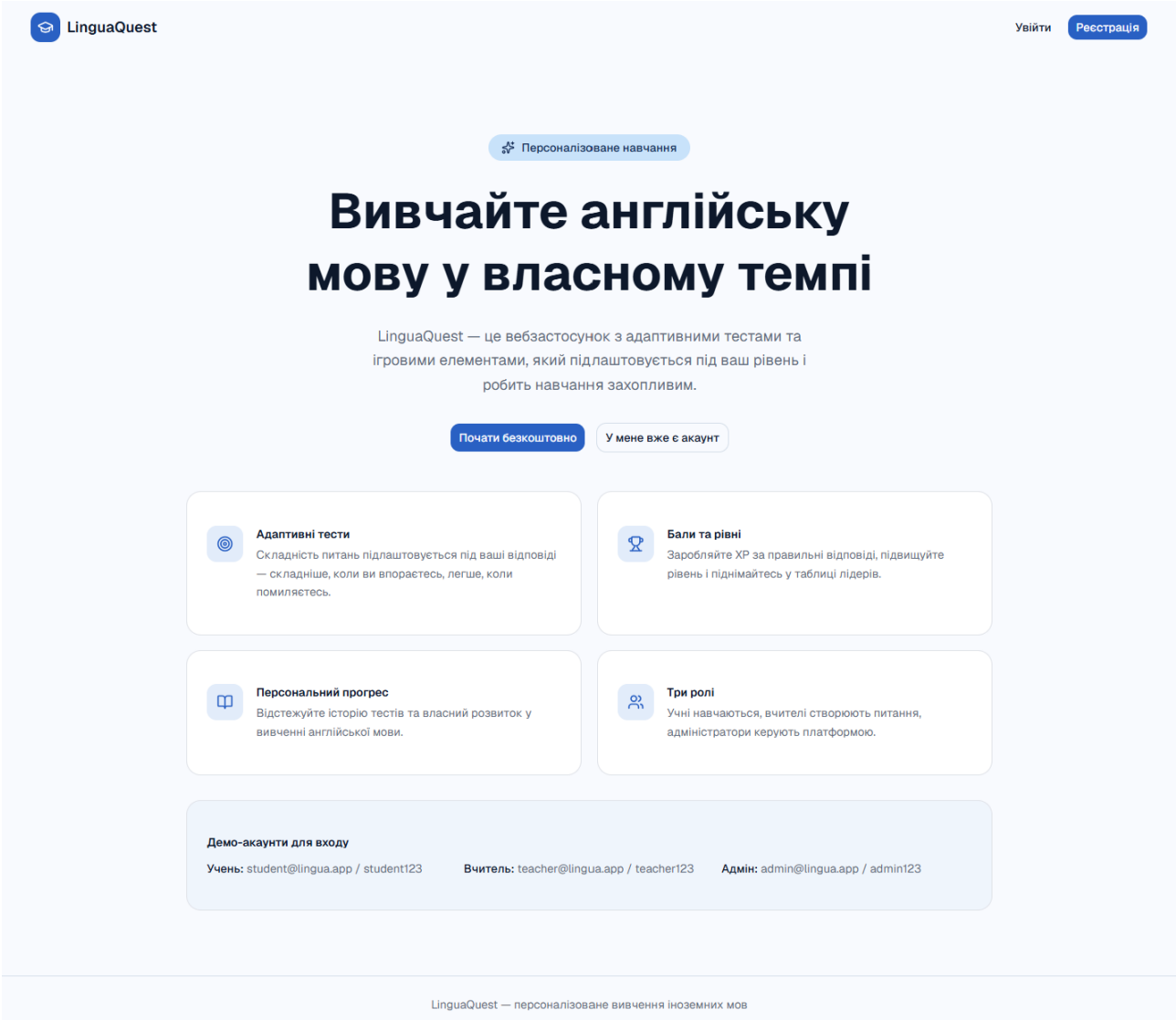


Рисунок Б.1 – Головна сторінка

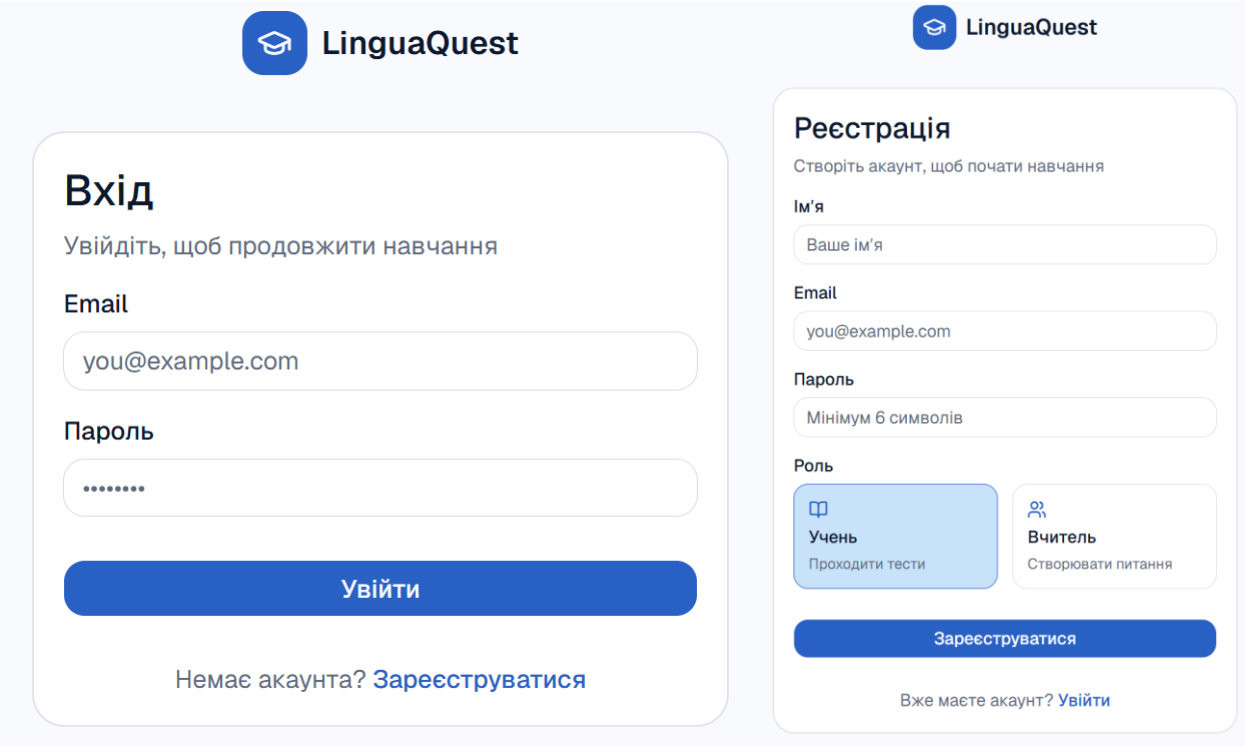


Рисунок Б.2 – Сторінка входу та реєстрації

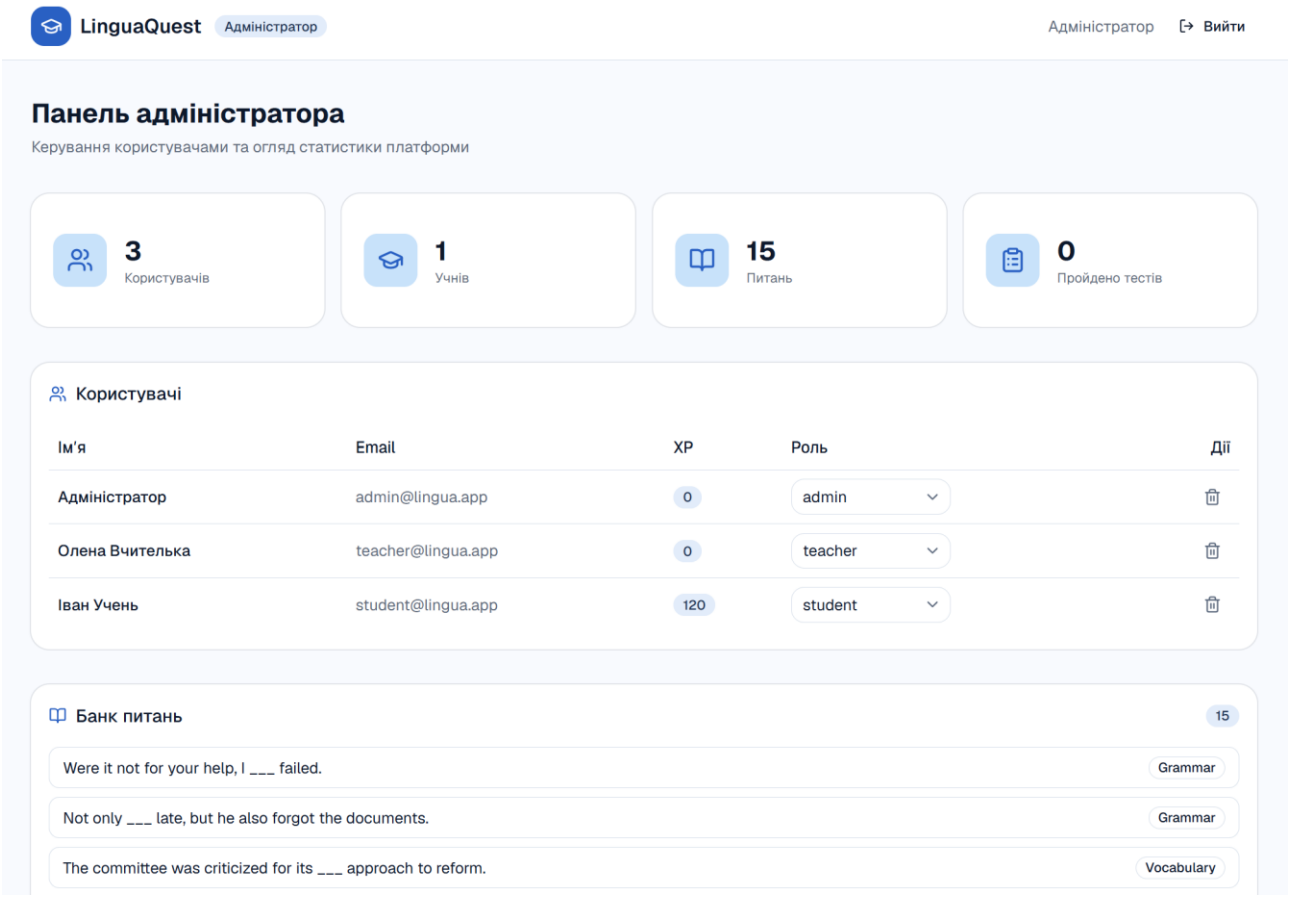


Рисунок Б.3 – Адмін-панель

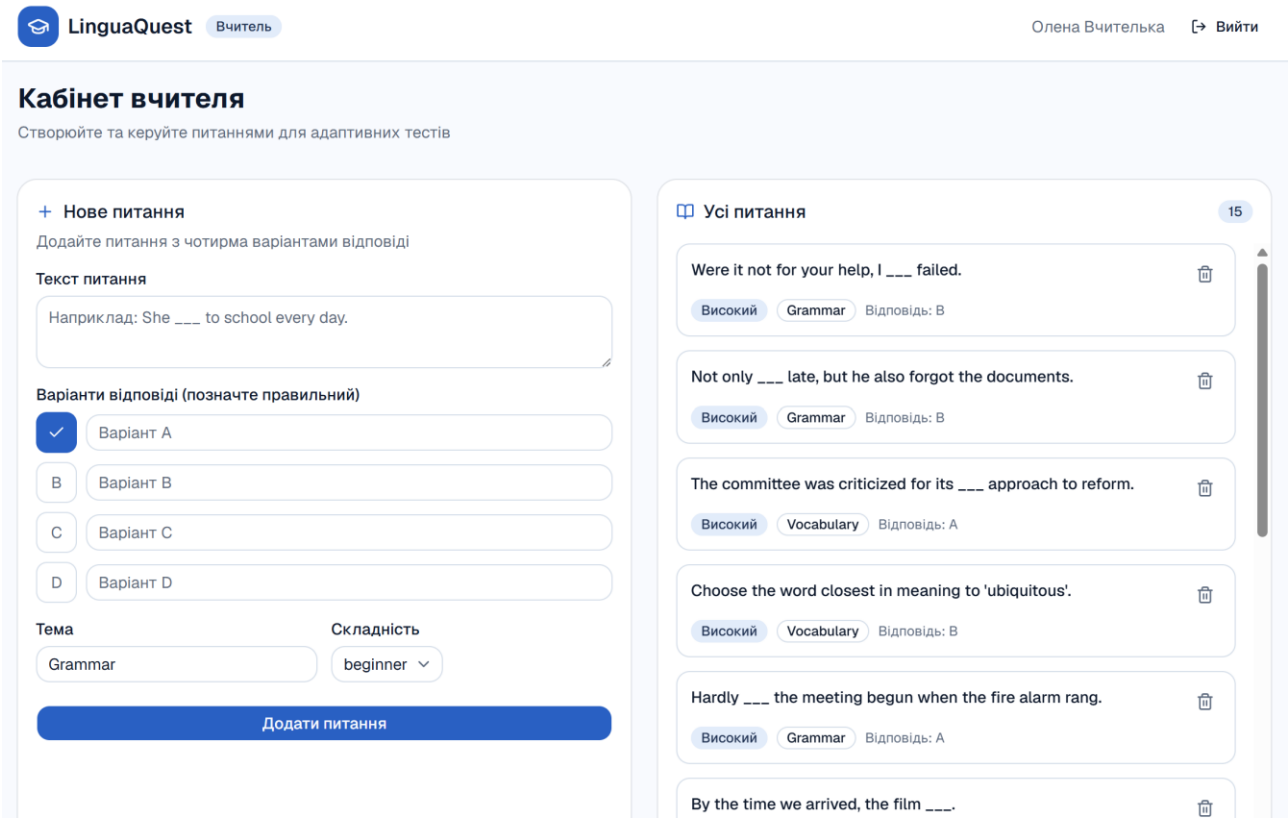


Рисунок Б.4 – Сторінка вчителя

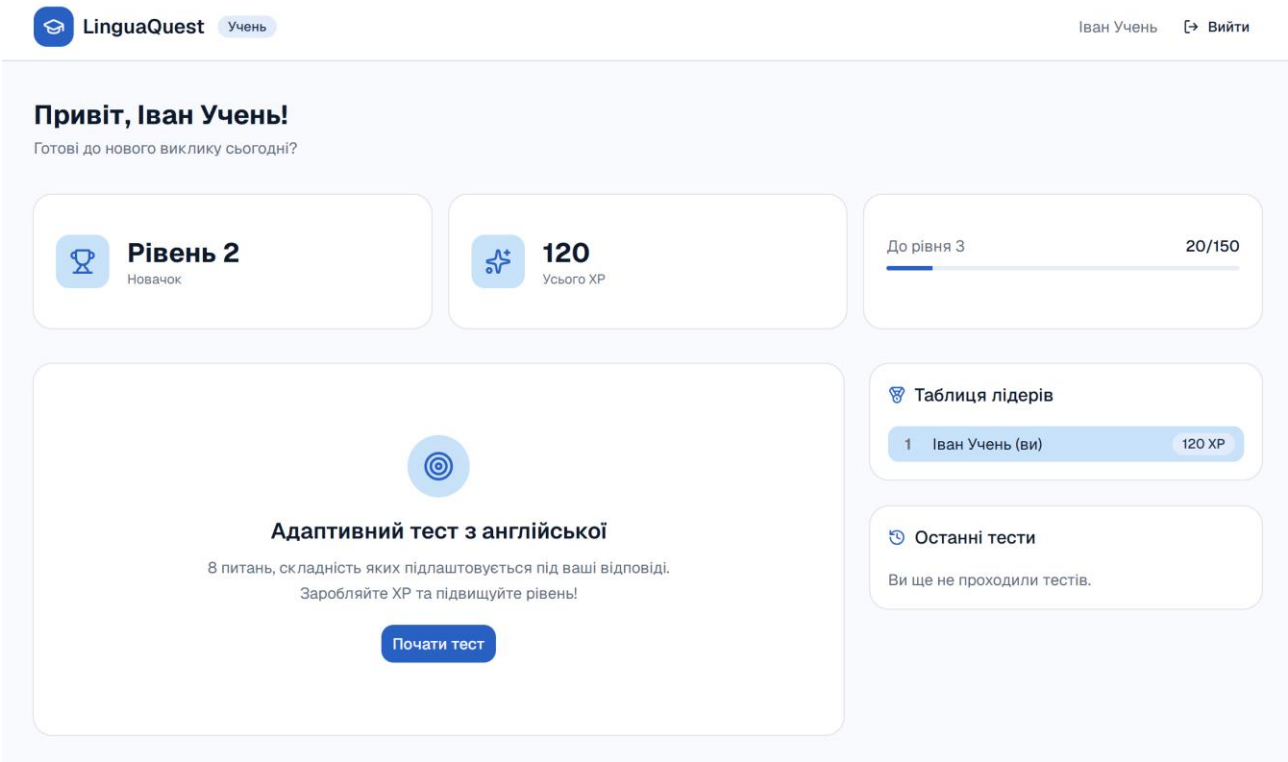


Рисунок Б.5 – Сторінка учня